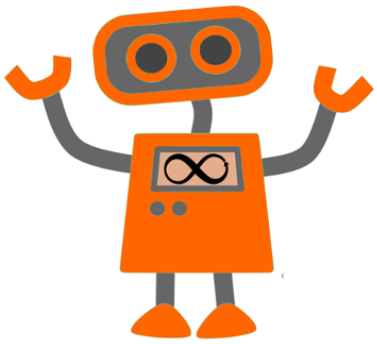
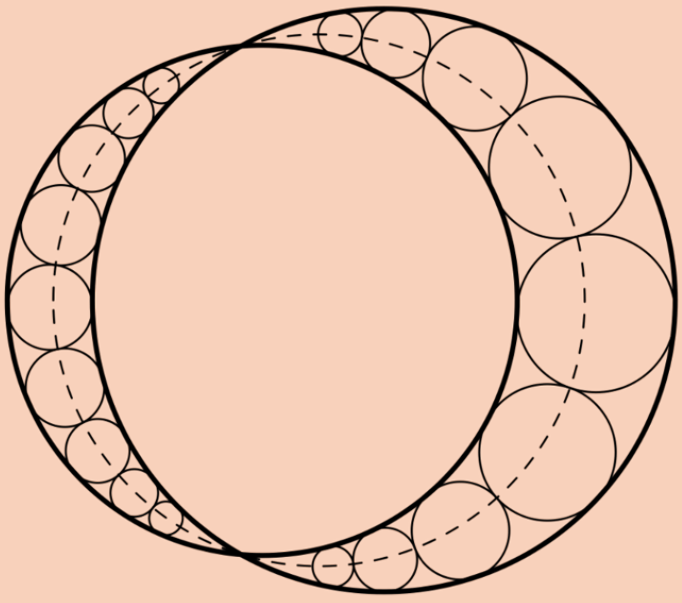
Mattekollo



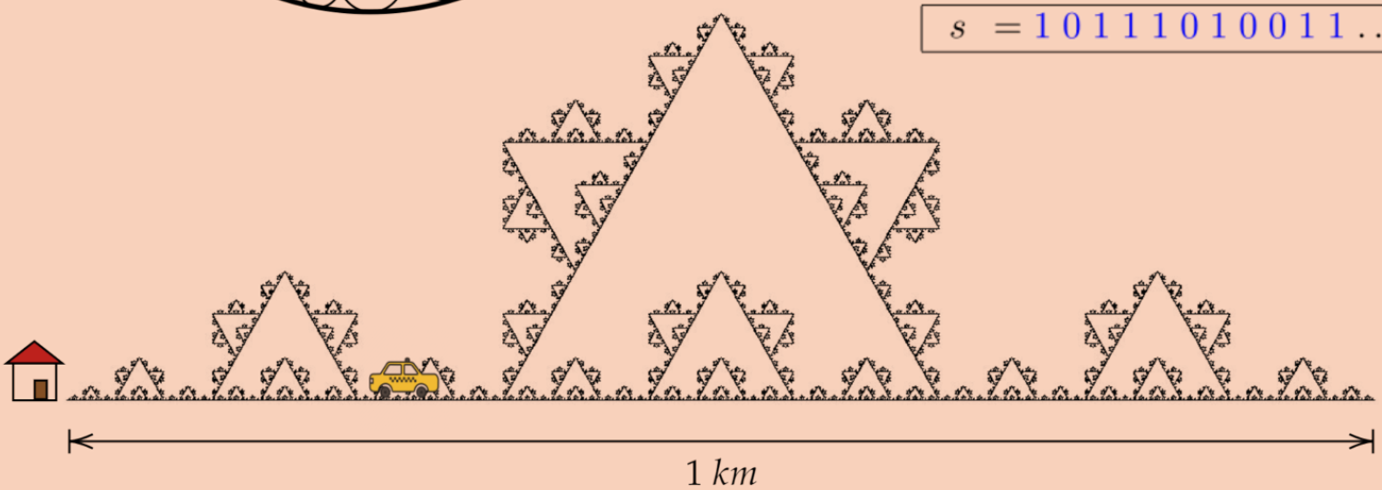
Lektionsmaterial

åk 6-gy2



$$
\begin{array}{ll}
s_1 & = 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\ldots \\
s_2 & = 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\ldots \\
s_3 & = 0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\ldots \\
s_4 & = 1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\ldots \\
s_5 & = 1\,1\,0\,1\,0\,1\,1\,0\,1\,0\,1\ldots \\
s_6 & = 0\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0\ldots \\
s_7 & = 1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\ldots \\
s_8 & = 0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\ldots \\
s_9 & = 1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\ldots \\
s_{10} & = 1\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\ldots \\
s_{11} & = 1\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0\ldots \\
\vdots & \quad \vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\ddots
\end{array}
$$

$$
s = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,1\ldots
$$



Mattekollo 2022

Lektionsmaterial

L. Andersson, V. Chapovalova, C. Ekbäck, H. Ekinge, D. Emanuelsson,

T. Glimmerfors, O. Kraft, E. Lundell, F. Löfgren, B. Verbeek och A. Villaro Krüger

Linköping, juni/juli 2022

Innehåll

Matematik - Grön

1. Talteori I: Delbarhet och Primtal

28 juni

Delbarhet

Definition. Vi säger att ett heltal b är ***delbart*** med ett heltal a om det existerar ett heltal k så att $ak = b$, och skriver $a|b$. Man kan även säga att a delar b, att b är en multipel av a eller att a är en delare till b.

▪ **Exempel 1** 42 är delbart med 6, eftersom $42 = 7 \cdot 6$. Det motsvarar $b = 42$, $a = 6$ och $k = 7$ i definitionen. ▪

1. Vilka delare har de här talen?

   (a) 42

   (b) 256

2. Vilka av dessa är multiplar av 15?

   (a) 5

   (b) 45

   (c) 751565

3. Det finns fyra påståenden om ett tal: Talet är delbart med 2, Talet är delbart med 3, Talet är delbart med 6, Talet är delbart med 12. Tre av dem är sanna, medan ett är falskt. Vilket?

4. **Lös i grupp:** Lös följande delbarhetsproblem:

   (a) Talet $a+1$ är delbart med 3. Visa att talet $4+7a$ också är delbart med 3.

   (b) Talen $2+a$ och $35-b$ är delbara med 11. Visa att talet $a+b$ också är delbart med 11.

5. **Extra:** Visa att om summan av vilka tre tal som helst utav fyra tal är delbar med 4, så är summan av alla fyra talen delbar med 4.

6. **Extra:** Bestäm det största fyrsiffriga talet som är delbart med 7 och som skrivs med fyra olika siffror.

7. **Extra:** Talet $14x+13y$ är delbart med 11 (för några heltal x och y). Visa att talet $19x+9y$ också är delbart med 11 (för samma x och y).

Primtal

Definition. Vi säger att ett heltal $p > 1$ är ett primtal om det bara delas av 1 och sig självt. Vi säger att ett heltal $n > 1$ är sammansatt om det inte är ett primtal, alltså om $n = ab$ för två heltal a och b som båda är större än 1.

■ **Exempel 2** 13 är ett primtal, eftersom de enda delarna till 13 är 1 och 13. 14 är ett sammansatt tal, eftersom $14 = 7 \cdot 2$. Det motsvarar $n = 14$, $a = 2$ och $b = 7$ i definitionen. ■

8. Hitta alla primtal mindre än 100. Hur många finns det?
9. Avgör om följande tal är primtal:

   (a) 91

   (b) 103

   (c) 143

   (d) 347

   (e) 12345

   (f) 387878

   (g) 1437004797

   (h) 3599

10. Talen 3, 5, 7 är alla primtal. Händer det någonsin igen att tre tal på formen n, $n+2$, $n+4$ alla är primtal?
11. Hitta alla positiva tal n så att de tre talen $3n-4$, $4n-5$ och $5n-3$ alla är primtal.
12. Hitta det minsta tresiffriga primtalet där varje siffra är ett primtal.
13. Mellan 10 och 20 finns det 4 primtal. Händer det någonsin igen att fyra tal mellan två på varandra följande multiplar av 10 är primtal?
14. Hitta 100 på varandra följande positiva heltal som alla är sammansatta.
15. Hitta det största tresiffriga primtalet.
16. **Extra:** Bevisa att det finns oändligt många primtal.
17. **Extra:** Bevisa att alla naturliga tal n > 1 kan skrivas som en produkt av primtal på minst ett sätt.

2. Talteori II: Aritmetikens fundamentalsats

29 juni

Faktorisering av tal

Sats 2.1. Varje ***heltal större än 1*** kan skrivas som en produkt av primtal. Detta kan göras på exakt ett sätt om man inte bryr sig om ordningen på faktorerna.

▪ **Exempel 3** Några primtalsfaktoriseringar:

- $12 = 2^2 \cdot 3$
- $360 = 2^3 \cdot 3^2 \cdot 5$

▪

1. **Lös i helgrupp:** En jägare berättade för sin kompis att han hade sett en varg i skogen som hade en 1 m lång svans. Kompisen berättade nyheten vidare och då sade att vargen hade en 2 m lång svans. När nyheten spreds vidare multiplicerade praktiska personer svanslängden med 3, medan teoretiska multiplicerade den med 2. Till slut sades det på TV-nyheterna att man hade sett en varg med 864 m lång svans. Hur många praktiska och hur många teoretiska personer fick vargens svans ”att växa”?

2. Benjamin har kommit på ett sätt att gömma flera tal i ett enda tal. Hans regel för att hitta det gömda talen är som följer: primtalsfaktorisera talet. Storleksordna primtalsfaktorerna. Första talet är antalet gånger första primtalsfaktorn förekommer, andra talet antalet gånger den andra primtalsfaktorn förekommer, och så vidare... Vilka tal, och i vilken ordning, har Benjamin gömt nedan?

   (a) 60

   (b) 105

   (c) 360

   (d) 2022

3. Skriv talet 111 som en summa av några heltal (inte nödvändigtvis olika) så att deras produkt också blir 111.

4. Det finns två kartongrektanglar med storlekarna 49 Œ 51 rutor och 99 Œ 101 rutor. Alla de delades upp i likadana rektanglar med heltalssidor. Rektanglarna var inte kvadrater. Bestäm storleken på rektanglarna.

5. Hur många 5:or finns det i primtalsfaktoriseringen av 729572743650000?

6. Bestäm ett tal som är delbart med 12 och som har 14 olika delare.

7. Kan ett tal som har exakt 15 delare vara delbart med (a) 100? (b) 1000?

8. Finns det något naturligt tal som vid multiplikation med 2 blir ett kvadrattal, vid multiplikation med 3 blir ett kubtal och vid multiplikation med 5 blir en femtepotens?

9. Kan man stryka bort en fakultets-multiplikationsterm från produkten $1! \cdot 2! \cdot 3! \cdot \cdots \cdot 100!$ så att det som står kvar blir ett kvadrattal?

3. Talteori III: SGD och MGM

30 juni

Största gemensamma delare - SGD

Definition. Den ***största gemensamma delaren*** av två (eller fler) heltal är det största heltal som är en delare till alla talen.

■ **Exempel 4** Ett par SGD:

- SGD(6, 33) = 3
- SGD(48, 180) = 12

■

1. **Lös i helgrupp:** Bestäm SGD av talen

   (a) 24 och 18

   (b) $313 \cdot 56 \cdot 132$ och $25 \cdot 37 \cdot 7 \cdot 113$

   (c) 2^{100} och 100^2

2. Hugo har räknat en svår uppgift och fått fram att svaret blir $\frac{5616}{8424}$. Nu vill han presentera lösningen snyggt och förkorta bråket till sin enklaste form. Hjälp honom! Vad blir svaret på den svåra uppgiften?

3. **Extra:** Det finns 6 stycken positiva heltal på tavlan. Man skriver upp alla möjliga SGD:er av ett par av dem på en separat tavlan. Kan talen på den andra tavlan vara just talen 1 till 15?

4. **Extra:** Ge exempel på (a) tre (b) fyra (c) hundra tal sådana att de inte har någon gemensam primtalsdelare allihopa, men varje par av dem har minst en gemensam primtalsdelare.

Definition. Två (eller fler) heltal sägs vara ***relativt prima*** om deras största gemensamma delare är 1.

5. Visa att n och $n+1$ är relativt prima.

6. **Extra:** Visa att på varandra följande Fibonaccital alltid är relativt prima.
   Ledtråd: om $a > b$ gäller att SGD(a, b) = SGD$(b, a-b)$

Minsta gemensamma multipel - MGM

Definition. Den ***minsta gemensamma multipeln*** av två (eller fler) heltal är det minsta heltal som är en multipel av alla talen.

■ **Exempel 5** Ett par MGM:

- MGM(6, 33) = 66
- MGM(48, 180) = 720

■

7. **Lös i helgrupp:** 3 elever på ett kollo är bra vänner och spelar spel tillsammans. Vid spelets slut bestämmer de sig för att samlas i matsalen igen senare för en ny omgång. Eleverna är dock rastlösa och om de andra 2 inte är där så går de en runda till innan de tittar efter igen nästa gång. Den ena eleven återkommer var 3:e minut, den andra var 8:e minut och den var 10:e minut. Hur lång tid dröjer det innan alla de 3 vännerna samlas igen?

8. Bestäm MGM av talen

   (a) 24 och 18

   (b) $313 \cdot 56 \cdot 132$ och $25 \cdot 37 \cdot 7 \cdot 113$

   (c) 2^{100} och 100^2

9. Bestäm ett par av positiva heltal om man vet att

   (a) deras SGD är lika med 56 medan deras MGM är lika med 112?

   (b) deras SGD är lika med 18 och deras MGM 630?

10. Visa att $\text{SGD}(a,b) \cdot \text{MGM}(a,b) = a \cdot b$

11. Lös följande kluringar:

    (a) Kan summan av två tal vara lika med deras minsta gemensamma multipel?

    (b) Kan summan av tre tal vara lika med deras minsta gemensamma multipel?

4. ”Omöjliga” konstruktioner

28 juni

Uppgifter för gruppdiskussion

1. I två påsar ligger två mynt, och dessutom ligger det dubbelt så många mynt i ena påsen som den andra. Hur är det möjligt?

2. Tre personer med en tvättmaskin vill åka över en flod. Deras båt rymmer antingen två människor och en tvättmaskin eller tre människor. Problemet är att tvättmaskinen är mycket tung och kräver tre människor som lastar den av och på båten. Kan personerna klara sitt mål?

3. På en rad står fyra tal, det första talet är lika med 100. Om man dividerar det första talet med det andra, det andra med det tredje, eller det tredje med det fjärde så blir kvoten alltid heltal som dessutom alla är primtal. Kan alla dessa tre primtal vara olika?

Uppgifter att lösa enskilt

4. Klipp ett hål i en A4-pappersark som du själv kan komma igenom.

5. Rita en sexhörning och genom att dra en linje genom två av dess hörn, ”klipp av” en sjuhörning från den.

6. Hur kan man med hjälp av sex tändstickor lägga ihop fyra trianglar som alla har tre sidor som är en tändsticka långa?

7. Hur kan man knyta en knut som på bilden genom att ta tag i ändarna på ett snöre och sedan inte få släppa dem förrän man knutit klart?

8. Kvoten utav två tal är lika med deras summa. Kan det även vara lika med talens produkt?

9. I en cirkel markerade man en punkt. Hur kan man klippa upp cirkeln i två delar som sedan kan byggas ihop till en cirkel med den markerade punkten i mitten?

10. Rita ett exempel på en sexhörning som inte kan delas upp i två fyrhörningar.

5. Ställa upp ett system

29 juni

Uppgifter för gruppdiskussion

1. Finns det något tresiffrigt tal som är 20 gånger större än sin egen siffersumma?

2. Bland fyra mynt är ett falskt och väger annorlunda än de andra (skulle kunna vara tyngre eller lättare, man vet inte), alla de riktiga mynten väger lika mycket. Hur kan man bestämma det falska myntet med hjälp av två vägningar på en balansvåg?

3. Trappan består av 130 trappsteg. På det nedersta trappsteget ligger 130 stenar, de andra trappstegen är tomma. På ett dygn får Sisyfos ta en grupp med stenar (en eller flera) från ett valfritt trappsteg och lägga dem på ett trappsteg som är lika många trappsteg bort som antalet stenar i gruppen man tog upp (dvs gruppen med en sten ska man lägga på ett trappsteg intill det man tog från, en grupp med två stenar — nästan intill osv.). Hjälp Sisyfos att lägga alla stenarna på trappsteget näst längst ner på max 15 dygn.

Uppgifter att lösa enskilt

4. Det finns ett sexsiffrigt nummer: 123556. Du får inte ändra ordningen på siffrorna. Sätt in räknetecken och eventuella parenteser mellan siffrorna, och kanske innan och efter, så att resultatet blir 100.

5. Bestäm åtminstone en lösning till rebusen $BRUTE + BRUTE = FORCE$. Olika bokstäver står för olika siffror, lika står för lika.

6. På en kvadrats gräns markerade man tre punkter, ritade ut de tre sträckorna mellan dem och klippte upp kvadraten längs med sträckorna. Man fick fyra bitar som alla var trianglar. Vilket är det största antalet sinsemellan likadana triangelbitar bland dessa?

7. Kan man placera ut en dam, två hästar och två torn på ett 4×5-bräde så att varje pjäs hotar exakt en annan pjäs och blir hotad av exakt en pjäs?

8. Sigge är yngre än Denise, men år 2021 fyllde båda de lika många år som siffersumman på deras egna födelseår. Vilka årtal är de födda?

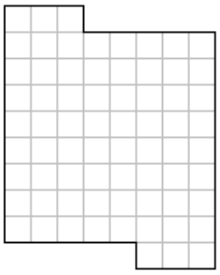
9. Finns det något tiosiffrigt tal där den första siffran från vänster är lika med antalet ettor i talet, den andra siffran från vänster är lika med antalet tvåor, den tredje med antalet treor, ..., den nionde med antalet nior, och den tionde med antalet nollor?

10. Genom att skriva talen $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots, \frac{1}{10}$ i någon ordning och skriva in räknetecken mellan dem (de tillåtna är de fyra vanliga räknesätten), få ett uttryck vars värde är lika med 0.

6. Flaskhalsprincipen

30 juni

Uppgifter för gruppdiskussion



1. På hur många sätt kan figuren delas upp i rektanglar av formatet

   (a) 1x5?

   (b) 1x7?

2. Valentinas bror ställer in datumet varje dag med hjälp av fem träblock.

Tre av dem är av avlånga och används för månaderna.

De andra två är kuber. Vilka siffror kan stå på kubernas sidor för att man ska kunna bilda alla datum?

3. Kan en kvadrat klippas upp i

   (a) en 30-hörning och fem femhörningar?

   (b) en 33-hörning och tre tiohörningar?

Uppgifter att lösa enskilt

4. Lös rebusen:

$$I + HE + HE + HE + HE + HE + HE + HE + HE = US$$

Som vanligt står likadana siffror för likadana bokstäver och olika står för olika.

5. Hur kan man plantera nio träd så att det bildas exakt 10 rader med tre träd i varje?

6. (a) Kan en schackhäst besöka alla rutor på ett 4x4-bräde utom en hörnruta exakt en gång?

(b) Kan man göra så fast man besöker alla rutorna exakt en gång?

7. (a) Kan talet 2021 skrivas som en summa av fem tal så att alla siffrorna i termerna man skriver upp är olika?

(b) Och som en summa av sex tal med samma villkor?

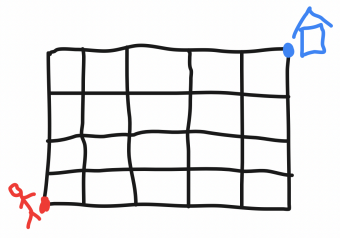
8. Elliot skrev upp 10 på varandra följande naturliga tal, bytte plats på dem på något sätt och skrev upp siffrorna utan mellanslag och fick då talet E. Tage skrev upp 11 på varandra följande naturliga tal, kastade om dem och skrev sedan upp talraden utan mellanslag och fick talet T. Kan $E = T$?

9. Vilket är det minsta antalet papperstrianglar som behövs för att slå in en kub utan vare sig hål eller överlapp?

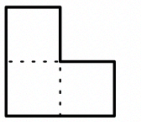
7. Combinatorics I

2 juli

1. In how many ways can we put 7 pearls on a wire such that no two pearls next to each other have the same colour? We can use pearls of 3 colours?



2. In how many ways can a $2\times$ 15 board be filled with L-shaped pieces?



3. Valentina has 5 different kinds of light chocolate and 5, and 5 different kinds of dark chocolate. In how many ways can she eat the chocolate, if she wants to eat dark and light chocolate alternatingly?

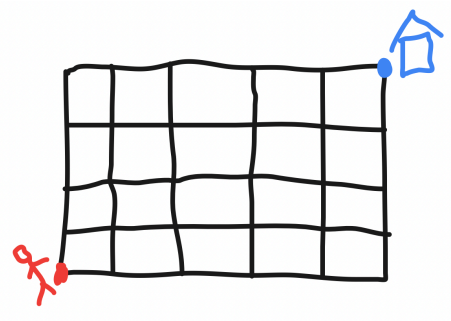
4.

- In how many ways can you pick a group of 2 students among 10 students?
- In how many ways can you pick a group of 3 students among 10 students?
- Can you make a general formula? In how many ways can you make a group of m students among n students?

Definition. Let m and n be integers such that $m \leq n$. We denote the number of ways to pick m elements among n different elements by $\binom{n}{m}$

5. $\binom{n}{m} = \binom{n}{m-n}$
6. (*) $1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{n-1} + \binom{n}{n} = 2^n$

7. In how many ways can Benjamin walk home following the straight lines, if he will take the shortest path?



8. (*) An ant is in a corner on a rubix cube. He wants to crawl the shortest way to the opposite corner, following the edges of the rubix cube. In how many ways may he do this? How can you generalize this to arbitrary sizes of the cube, and to higher dimensions?

9. There are 10 horses standing in a row. Because of a dangerous virus, the farmer has decided to built 3 walls between the horses. In how many ways may the farmer do this?

10. 4 robbers have stolen 10 (identical) horses. In how many different ways can they divide the horses between them? (Some robbers may receive 0 horses).

11. In how many ways can you choose 5 positive integers x_1, x_2, x_3, x_4, x_5 such that their sum is 25?

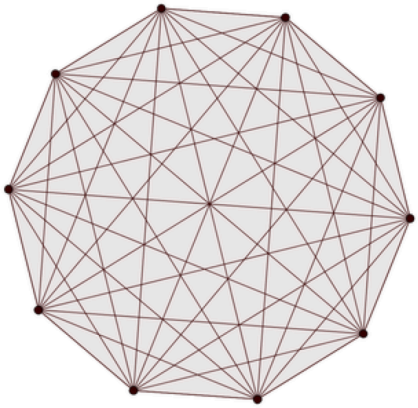
12. Come up with a general formula: In how many ways can you divide m (identical) horses among n robbers?

13. $1 \cdot \binom{n}{1} + 2 \cdot \binom{n}{2} + 3 \cdot \binom{n}{3} + \ldots + (n-1) \cdot \binom{n}{n-1} + n \cdot \binom{n}{n} = n \cdot 2^{n-1}$.

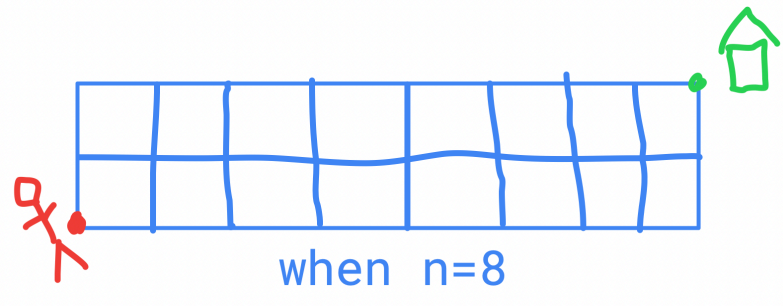
8. Combinatorics II

3 juli

1. How many diagonal lines are there in an $10-$gon? How many are there in an $n-$gon?



2. To walk home, benjamin will walk two times up, and n times right (the shortest path). In how many different ways can he do this?



3. There are 200 different pieces of candy in a box. In how many ways can you pick 198 of the pieces?

4. En klasse med 20 elever skal vælge en formand, en næstformand og en kasserer. På hvor mange måder kan dette gøres?

5. En klasse med 20 elever skal vælge 3 elever til elevrådet. På hvor mange måder kan dette gøres?

6. (*) In how many ways can you place k towers on a $n \times n$ chessboard, such that they do not threaten each other?

7. There are 10 horses standing in a row. Because of a dangerous virus, the farmer has decided to built 3 walls between the horses. In how many ways may the

farmer do this?

8. 4 robbers have stolen 10 (identical) horses. In how many different ways can they divide the horses between them? (Some robbers may receive 0 horses).

9. In how many ways can you choose 5 positive integers x_1, x_2, x_3, x_4, x_5 such that their sum is 25?

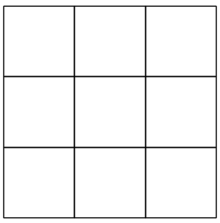
10. Come up with a general formula: In how many ways can you divide m (identical) horses among n robbers?

11. (*) A box contains a lot of socks. The socks are either red or blue. The total number of socks is not greater than 1993. If you pick two arbitrary socks from the box, the possibility that they have the same colour is $\frac{1}{2}$. Given this information, what is the largest possible number of red socks?

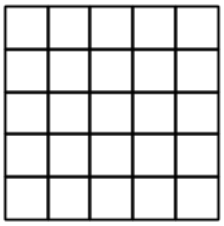
9. Kombinatorik 3: Visuella uppgifter

4 juli

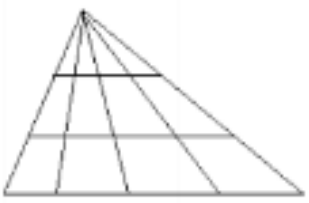
1. Hur många kvadrater kan man se på ritningen?
   (a)



   (b)



2. Hur många trianglar kan man se på bilden?



3. På en cirkel finns 5 röda, 7 blå och 9 gröna punkter. Hur många olika trianglar går det att skapa med hjälp av punkterna om
   (a) alla hörnen måste vara gröna?
   (b) alla hörnen måste vara av samma färg?
   (c) alla hörnen måste vara av olika färger?
   (d) inte alla hörnen ska vara av samma färg?

4. På ett rutat papper finns en 6x10-rektangel. På hur många sätt går det att markera en rektangel vars sidor går längs med rutnätets linjer?

5. Hur många olika delrätblock finns det i en $n \times m \times k$-rätblock?

10. Logik I: Påståenden och slutledningar

2 juli

Definition. Ett ***påstående*** är någonting som kan vara antingen sant eller falskt. Dessa finns i flera olika former, bland annat som slutledningar.

Definition. Logiskt ***och*** betyder att alla påståenden som hör till måste vara sanna för att påståendet som helhet ska vara sant.

Definition. Logiskt ***eller*** betyder att åtminstone ett av påståenden som hör till måste vara sanna för att påståendet som helhet ska vara sant.

Definition. Logisk ***negation*** ger ett påstående som är tvärtom det ursprungliga. Det är sant när det ursprungliga påståendet är falskt och vice versa.

1. Vilka av meningarna är påståenden?

   (a) 5

   (b) Jag är en människa

   (c) Solen är en stor gul apelsin.

   (d) Programmering är bäst.

   (e) Matematik, kollo och plustecken.

2. Vilka av påståenden nedan är sanna?

   (a) Just nu är det sommar och Hugo är en jätte.

   (b) Just nu är det vinter eller Hugo är en människa.

   (c) Just nu är det sommar eller Benjamin är en människa.

3. Formulera negationer (matematiska motsatser) till följande meningar:

   (a) Vägen svänger åt vänster.

   (b) Talet x är mindre än 17.

   (c) Findus är fågel eller fisk eller mittemellan.

   (d) Det är soligt.

   (e) $5+2=7$

4. Formulera ett påstående som

(a) Alltid är sant.

(b) Alltid är falskt.

(c) Inte går att veta om det är sant eller falskt.

5. Vilka av dessa slutledningar är sanna?

(a) Jag har en fotboll. Alla fotbollar är runda. Alltså har jag minst en rund sak.

(b) Varje gång jag får alla rätt på ett prov äter jag glass. Idag åt jag glass, alltså har jag fått alla rätt på ett prov.

(c) Jag har en liten och grön katt, alla starka katter är gröna. Alltså är min katt stark.

6. Vilka av påståenden nedan är sanna?

(a) Om Usain Bolt har sprungit snabbast i världen så har han ett världsrekord.

(b) Om Usain Bolt har sprungit snabbast i världen så är Jupiter en planet.

(c) Om $5 > 11$ så är Jupiter en planet.

(d) Om $5 > 11$ så ligger Linköping i Indien.

(e) Om Usain Bolt har sprungit snabbast i världen så ligger Linköping i Indien.

7. **Extra:** Betrakta meningen: ”Om denna mening är sann så existerar jultomten”.

(A) Är meningen sann?

(B) Existerar jultomten?

(C) Varför är meningen konstig?

8. **Extra:** En mattekollodeltagare vet inte riktigt var busshållplatsen ligger. Hen vet att den antingen är precis utanför Ormbunken eller vid matsalen eller på andra sidan fotbollsplanen. Hen frågar kolloledarna A, B, C, D, E om råd. Problemet är att vissa av ledarna alltid ljuger (medan andra alltid talar sanning), men eleven vet inte vem som är av vilken sort. Följande råd gavs:

(A) Busshållplatsen är precis utanför Ormbunken!

(B) Klart att hållplatsen ligger vid matsalen!

(C) Det är inte sant att både A och B ljuger hela tiden!

(D) Jag lovar att A ljuger eller B talar sanning!

(E) Lyssna på mig: Jag ljuger eller så är C och D av samma sort (båda lögnare eller båda sanningssägare).

11. Logik II: Mängdlära

3 juli

1. Rita två mängder: Personer som spelar Minecraft (M) och personer som spelar League of Legends (L). Markera sedan följande på var sin bild:

   (a) Personer som spelar Minecraft och LoL ($M \cap L$).

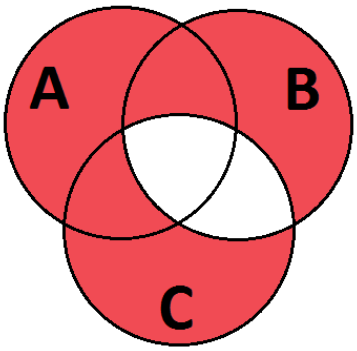
   (b) Personer som spelar Minecraft eller LoL ($M \cup L$).

   (c) Personer som spelar Minecraft, men inte LoL ($M \setminus L$).

   (d) Personer som inte spelar Minecraft ($\overline{M}$).

   (e) Personer som spelar exakt ett av spelen. Hur skulle man kunna beteckna den delen?

2. Tre mängder skär varandra så som bilden visar. Hur kan man uttrycka den röda delen med hjälp av bokstäverna A, B, C samt symbolerna $\cup, \cap, \setminus$?



3. Vilka av dessa likheter är sanna?

   (a) $\overline{A \cup B} = \overline{A} \cup \overline{B}$

   (b) $\overline{A \cup B} = \overline{A} \cap \overline{B}$

   (c) $\overline{A \cap B} = \overline{A} \cup \overline{B}$

   (d) $\overline{A \cap B} = \overline{A} \cap \overline{B}$

4. Vilka av dessa uttryck betecknar samma mängd?

   (a) $B \cap C \cap \overline{A}$

(b) $B \cap C \cap A$

(c) $A \cup (B \cap C)$

(d) $(B \setminus A) \cap (C \setminus A)$

(e) $(C \setminus B) \cup (C \setminus A)$

5. Är det sant att $A \cap (B \setminus C) = (A \cap B) \setminus (A \cap C)$ och $A \cup (B \setminus C) = (A \cup B) \setminus (A \cup C)$ gäller för alla mängder A, B och C? Bevisa eller ge motexempel!

6. På ett dagis fanns massa kort för att lära sig att läsa, på några står det ”HA” och på några står det ”KA”. Varje barn fick tre kort. Det visade sig att 20 barn kunde bygga ordet ”HAHA” med sina kort, 30 barn kunde bygga ”KAKA” och 40 barn kunde bygga ”HAKA”. Hur många av barnen hade tre likadana kort?

7. **Extra:** Alex, Benjamin och Clara skrev var sin lista på filmer de hade sett det senaste året, totalt hade de sett 25 olika filmer tillsammans. Alexs lista blev 16 filmtitlar lång, Benjamin hade sett 17 och Clara hade sett 9 filmer. Några filmtitlar fanns då förstås med på flera listor: Alex och Benjamin hade 10 filmtitlar som deras listor hade gemensamt, medan Alex och Clara hade 4 gemensamma filmtitlar. Hur många filmer såg Benjamin som ingen annan av de andra två hade sett?

8. **Extra:** Victor, Ludvig och Hanna löstes 100 matteproblem tillsammans. Var och en av dem löste 60 problem. Låt oss säga att ett problem var svårt om endast en person löste det och lätt om alla tre löste det. Hur mycket skiljer sig antalet svåra problem från antalet lätta problem?

9. **Extra:** (från HMT-1988) Samtliga elever vid en skola tillfrågades om sina sportaktiviteter. Det visade sig att 90% av eleverna spelade boll i någon form, 80% sysslade med friidrott, 70% åkte skidor och 60% åkte skridskor. Ingen elev ägnade sig åt alla dessa fyra aktiviteter. Hur många elever åkte skidor eller skridskor eller bäggedera?

12. Logik III: Binära tal

4 juli

Definition. ***Binnära tal*** är tal som är skrivna i basen 2. Det betyder att varje siffra i talet endast kan anta värdena 0 eller 1, samt att varje position är värd en potens av 2 istället för en potens av 10 som vi är vana vid när vi räknar ”som vanligt”.

■ **Exempel 6** $39_{10} = 3 \cdot 10^4 + 9 \cdot 10^0$

$$100111_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 32 + 0 + 0 + 4 + 2 + 1$$

■

1. Konvertera följande binära tal till tal med bas 10:

   (a) 110010101_2

   (b) 1001001_2

   (c) -1110101100_2

   (d) -100010_2

2. Konvertera följande tal med bas 10 till binära tal:

   (a) 382_{10}

   (b) 45_{10}

   (c) -227_{10}

3. Beräkna följande operationer på två sätt, både med hjälp av och utan konvertering till tal med bas 10:

   (a) $1010011001_2 + 100101001010_2$

   (b) $101011011_2 + 111111111_2$

   (c) $111001011_2 - 101001011_2$

   (d) $110101_2 - 111010_2$

4. Beräkna följande operationer på två sätt, både med hjälp av och utan konvertering till tal med bas 10:

   (a) $10101010101_2 \cdot 1010101101_2$

   (b) $110100000_2 \cdot 11110000000_2$

5. Vad händer med ett binärt tal om man

(a) fördubblar talet?

(b) halverar talet?

(c) multiplicerar talet med 8_{10}?

6. Beräkna resultatet **utan** att omvandla talen till decimala talsystemet:

(a) $10000_2 \cdot 100_2$

(b) $10101_2 \cdot 1010_2$

(c) $11111_2 \cdot 1_2$

7. Anders tänker på ett tal mellan 1 och 1000. Man får ställa ”ja/nej”-frågor till honom för att gissa talet. Kan man garanterat gissa hans tal på 10 frågor?

8. Dela upp en 7×7-kvadrat i nio rektanglar (inte nödvändigtvis olika), så att man med hjälp av dem skulle kunna bygga vilken rektangel som helst, så länge den har sidor som är 7 eller mindre. (Här menar vi alltid rektanglar, vars sidor mäts i ett helt antal rutor.)

9. **Extra:** Du sitter i en fängelsehåla och en vakt kommer till dig med ett erbjudande. Han säger att han kommer lägga upp mynt (totalt 64 st.) på ett schackbräde, ett mynt i varje ruta och det kommer vara slumpat för varje mynt huruvida det ligger med krona eller klave uppåt. Därefter kommer vakten peka på en ruta. Du får sedan en möjlighet att vända upp-och-ner på exakt ett av mynten, vilket du vill.

Därefter kommer en kompis in i rummet, som får se schackbrädet och får gissa vilken ruta vakten pekade på. Gissar han rätt, får ni båda gå fria. Hur kan du och kompisen komma överens om en strategi som garanterar er frihet, oavsett hur vakten gör?

13. Logik IV: Logiska kretsar

5 juli

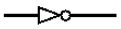
1. Om falskt är 0 och sant är 1 (och alla andra tal som är större än 1), så kan vi räkna med sant och falskt som med binära tal:

$$\begin{aligned} 0+0&=0 & 0\cdot 0&=0\\ 0+1&=1 & 0\cdot 1&=0\\ 1+0&=1 & 1\cdot 0&=0\\ 1+1\ (=10)&=1 & 1\cdot 1&=1 \end{aligned}$$

Vilken av $+$ och $\cdot$ är matematiskt "och" och vilket är matematiskt "eller"?

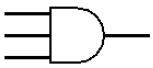
■ **Exempel 7** Nedanstående figur visar hur man kan beskriva logiska operationer med hjälp av elekriska kretsar: ■

Inte (byter värde)



Eller (sann om ett ingående värde är sant)

Och (sann om alla ingående värden är sanna)



Antingen eller (sann om exakt ett av ingående värdena är sann)

2. Beräkna vad kretsen nedan genererar om:

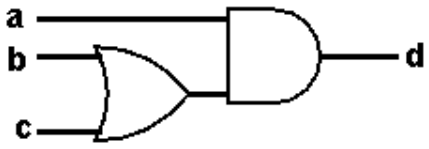
(a) $a=1$, $b=0$ och $c=1$

(b) $a=1$, $b=1$ och $c=0$

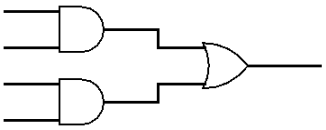
(c) $a=0$, $b=1$ och $c=1$

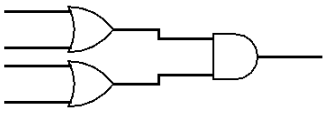
(d) $a=0$, $b=0$ och $c=1$

3. Gör en sannningstabell för nedanstående krets:

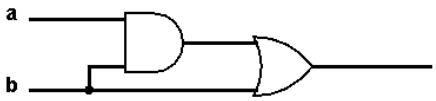


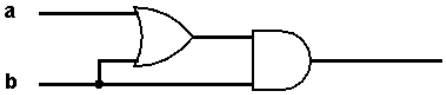
4. Förklara hur man kan generera värdet 1 i nedanstående kretsar:



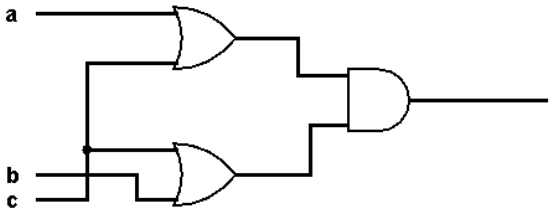


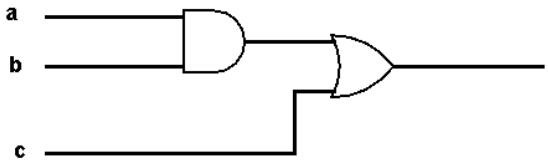
5. Undersök vad de nedanstående kretsarna genererar:





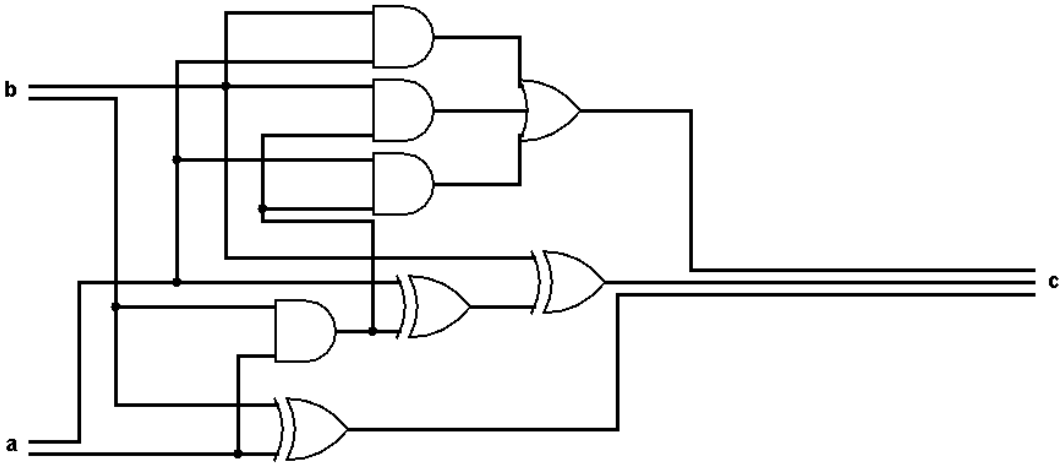
6. Visa att de nedanstående kretsarna fungerar likadant:





7. **Extra:** Rita kretsen som fungerar precis som uttrycket $a \cdot b + a \cdot c$. *Ledning: När uttrycket är skriven på den formen tar man alltid "multiplikationen" först. Tag hjälp av uppgift 1.*

8. **Extra:** Låt a och b vara 2-siffriga binära tal, och låt c vara ett tresiffrigt binärt tal. Beräkna c för några värden på a och b i den nedanstående kretsen. Kan du komma på hur kretsen fungerar?



9. **Extra: (1-bit adderare)** Svante tröttnade på att göra uppställningar för att addera binära tal med varandra. Han vill därför istället bygga en krets av logiska grindar som gör jobbet åt honom. Hjälp honom med detta problem!

Kretsen ska alltså ta in två binära tal a och b som är 1 bit stora, och ge resultatet c enligt $c = a + b$. Kretsen ska även ge ut en så kallad carry-bit som vi betecknar med c_{out}. För att vara riktigt användbar ska kretsen även ta in en carry-bit från en tidigare addition som vi kallar c_{in}.

Ledning: Detta går att göra på flera olika sätt, men det är möjligt att lösa uppgiften med endast 5 grindar om man blandar "antingen eller", "och" och "eller"-grindar.

10. **Extra: (4-bits adderare)** Kretsen från uppgiften innan är fortfarande inte tillräckligt användbar tycker Svante. Hjälp honom att generalisera genom att bygga ihop flera block så att din nya krets kan ta emot 2 st 4-bitarstal och summera dem till ett tal med upp till 5 bitar!

Ledning: Använd din lösning från uppgiften innan. Det är okej att hitta på en ny symbol som definieras som en samling av grindar som du bestämmer.

14. Erövringar gröna gruppen

5 juli

Regler. På tavlan finns en karta. Varje lag får en lista med problem. Problemen kan lösas i valfri ordning. På varje problem har man två försök att lämna in rätt svar (endast svar lämnas in). Om ett svar är rätt får laget ta ett valfritt ledigt område på kartan. Om ett lag vill ta över ett område som är upptaget måste man ha rätt på två (nya) problem. Laget som har flest områden i slutet vinner.

1. Bestäm alla möjliga lösningar till matematiska rebusen $J \cdot AHA = MEH$.

2. Sätt ut (alla) siffrorna 1, 2, 3, 4, 5, 6, 7, 8, 9 i en 3x3-tabell så att summan av talen i varje rad, varje kolumn och varje stordiagonal blir exakt samma.

3. Kan man fördela 99 nötter i 10 högar på så sätt att alla högarna innehåller olika många nötter men ingen hög är mer än tre gånger så stor som en annan? Om det går, visa hur man kan göra.

4. Dela upp en liksidig triangel i 7 mindre liksidiga trianglar (som inte behöver vara lika stora).

5. Under Mattekollo släppte kollokatten Matte följande på Valentinas matta: sex grå möss, tolv möss utan svans och femton möss som hen tuggat lite på.
Dessutom så var bara en mus grå, utan svans och tuggad på. Två var grå och svanslösa men hade inte blivit tuggade på, och så fanns det två möss som var grå och tuggade på, men hade kvar sin svans.
Om det totalt fanns 24 möss på Valentinas matta och alla var antingen grå, svanslösa eller tuggade, hur många var svanslösa och tuggade men inte grå?

6. Tusen hundar skakar tass med varandra. Hur många tasskaningar sker om varje hund skakar tass med varje annan hund?

7. Karin har bakat en ljuvlig kladdkaka och ställt i matsalen. Efter att städpatrullen har varit och städat märker hon dock att den har blivit är uppäten. Städpatrullen bestod av Mika, Elliot och Anders. När Karin frågar ut dem inser hon att 2 av dem ljuger och 1 talar sanning.
Mika: Elliot åt kladdkakan.
Elliot: Mika ljuger.
Anders: Jag åt inte kladdkakan.
Vem var det egentligen som åt kladdkakan?

8. Det finns sex trådar. Varje trådände paras ihop med en annan trådände sen knyts dessa ändar samman. Vad är sannolikheten att de bildar exakt en ring?

9. Alla heltal från 1 till 1000 skrev man på rad i följande ordning. Först skrev man alla tal som har siffersumma 1 i stigande ordning, sedan talen som har siffersumma 2 i stigande ordning, sedan siffersumma 3 på samma sätt, och så vidare. Vilket tal står på plats 996?

10. Två lika stora böcker skall läggas i en låda som är kubformad. När en bok läggs med sin största sida mot lådans botten får den precis plats till ytan. På hur många olika sätt kan böckerna läggas i lådan?

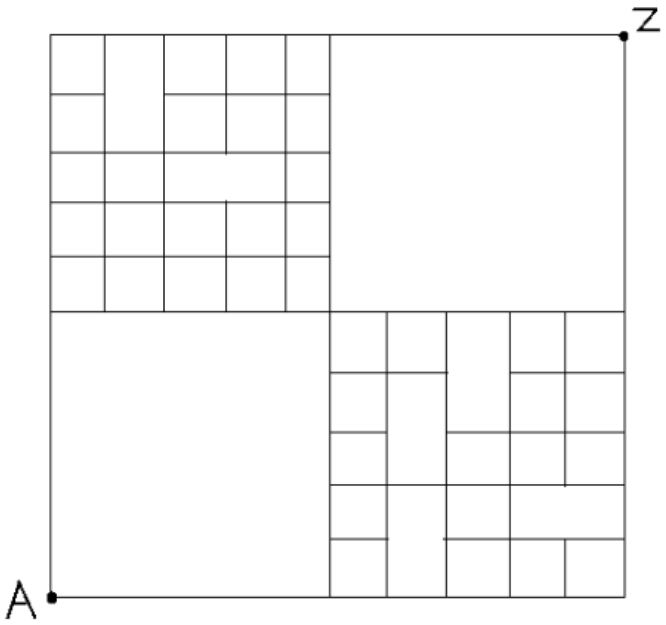
11. Vad ska stå under den sista raden?

1
1 1
2 1
1 2 1 1
1 1 1 2 2 1
3 1 2 2 1 1

12. Bestäm antalet positiva heltal som är mindre eller lika med 1000 och som inte är delbara med 3,4,5 eller 6.

13. Nio likadana fåglar äter mindre än 1001 frön, medan tio fåglar av samma sort äter mer än 1100 frön. Hur många frön äter varje fågel?

14. Bestäm antalet kortaste vägar från A till Z.



15. Hur många femsiffriga tal finns det där alla siffror är jämna?

16. Sätt ut räknetecken så att likheten blir korrekt:

$$9\ \ 9\ \ 9\ \ 9\ \ 9 = 11$$

17. Dela upp rektangeln 6x9 i åtta kvadrater.

18. I den mörka skogen har den Magiska Granen vuxit i över hundra år. Varje morgon växer det ut 1000 nya barr på den. Varje barr lever i ett exakt 1 år och dör sedan av. Hur många barr hade den Magiska Granen exakt klockan 12.00 idag?

II

Matematik - Röd

1. Invarianter I

28 juni

1. På tavlan så är ett antal trianglar och ett antal cirklar uppskrivna. Ett tillåtet drag är att sudda ut en cirkel och en triangel. Går det att sudda ut hela tavlan med hjälp av sådana drag om vi startar med 5 cirklar och 7 trianglar?

2. På tavlan står talen $1,2,3,4,5,6,7,8$. Erik väljer två tal slumpmässigt, suddar ut dem, och ersätter dem med sin summa. Kan han få talet 34? Vilka tal kan han få?

3. Kan schackbräde med 8×8 rutor täckas med dominobrickor (2×1 rutor) om vi tar bort två diagonalt motstående hörn?

4. På tavlan står återigen talen $1,2,3,4,5,6,7,8$. Erik väljer två tal, suddar ut dem, och ersätter dem med sin *positiva differens*. Kan summanav talen på tavlan någonsin bli 15?

5. Ett schackbräde är färgat som vanligt, med varannan ruta svart och varannan vit. I ett drag får man byta färg på alla rutor i en 2 Œ 2 ruta eller på alla rutor i en rad eller kolumn. Går det att byta färg på några rutor med dessa drag så att exakt en ruta är svart i slutet?

6. Anna och Björn har en chokladkaka som består av 7×10 rutor. Varje spelare får antingen bryta av en bit av chokladkakan längs en linje som separerar rutor (så 7×10 $3 \times 10 + 4 \times 10$) eller äta upp en sammanhängande bit. Den som äter sista biten förlorar. Anna börjar. Vem kommer att vinna?

7. Vi startar på en punkt med koordinater $(1,0,0)$ i rummet. Ett 3-dimensionellt springardrag ändrar en koordinat med 1, en annan koordinat med 2 och den kvarvarande koordinaten med 3. Exempelvis så kan vi ta oss från $(2,2,2)$ till $(3,-1,0)$. Kan vi ta oss från $(1,0,0)$ till $(0,0,0)$ med en följd av sådana springardrag?

8. Det finns 60000 personer i Karlstad varav 1 är en zombie. Varje dag så samlas alla personer slumpmässigt i grupper om 3. Om en grupp innehåller en zombie så blir alla tre personer i gruppen zombier. Vad är sannolikheten att det finns precis 100 zombier efter 1 vecka?

9. Betrakta talet 9999^{9999}. Tag dess siffersumma, och tag sedan siffersumman av det och fortsätt på samma sätt. Kan slutresultatet bli 7?

2. Invarianter II

29 juni

1. Du har precis träffat en drake med 100 huvuden, och den är arg! Med ett slag kan du hugga av antingen 5, 12, 17 eller 20 av huvudena, men då växer 17, 24, 2 respektive 17 nya ut. Om draken förlorar alla huvuden växer inga nya ut, den förlorar, och du blir en hjälte! Kan du besegra draken?

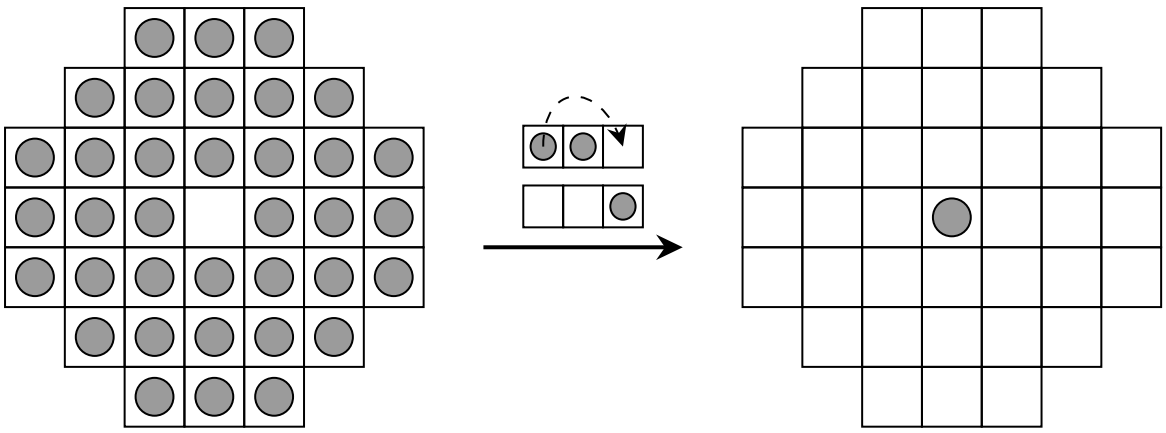
2. På en tavla står 50 tal mellan 1 och 100. Vi kan i ett steg byta ut två tal a och b mot talen $\frac{a+b}{2}$ och $\sqrt{ab}$. Efter att ha gjort detta massor av gånger räknar vi ut skillnaden mellan största och minsta talet som står på tavlan. Visa att skillnaden inte kan blir större än 99.

Lös i valfri ordning, gärna i grupp

3. Det finns tre högar med 5, 49 respektive 51 stenar. Ett drag består antingen av att man kombinerar två högar till en hög, eller av att man delar upp en hög med ett jämnt antal stenar i två lika stora högar. Är det möjligt att med ett ändligt antal drag dela upp stenarna bi 105 högar med en sten i varje?

4. Betrakta planet med positiva heltalskoordinater $\{(a,b) \mid a,b \in \mathbb{Z}^+\}$. Det ligger en sten i nedre vänstra hörnet på vardera $(1,1),(1,2)$ samt $(2,1)$. I ett drag får du ta bort en sten (x,y) och placera en ny sten ett steg upp på $(x,y+1)$, och en annan ett steg till höger på $(x+1,y)$, givet att de koordinaterna är tomma (annars får du inte ta bort stenen från (x,y)). Går det att med ett ändligt antal sådana drag att få bort alla stenar från de tre ursprungskoordinaterna?

5. Hugo spelar ”Europeisk solitär”. Brädet, som ser ut som bilden nedan, börjar fyllt av stenar, utom en i mitten. Genom att hoppa en sten över en annan, och sedan ta bort den sten som hoppades över, får man minska antalet stenar. Hugo försöker göra så att det endast blir en sten kvar. Visa att det är omöjligt.

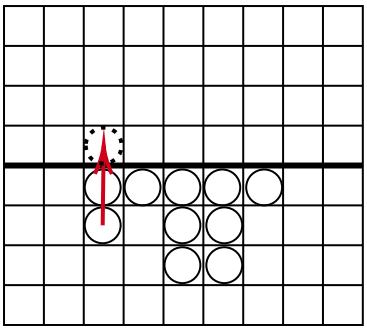


3. Invarianter III: Conway's soldiers

30 juli

Idag löser vi ett mer komplicerat invariantproblem:

På ett oändligt rutnät finns en lång rät linje dragen. Under linjen får vi i början placera så många stenar vi vill, men högst en sten per ruta. Därefter får vi lov att ta en sten och hoppa över en intilliggande sten, om rutan vi hoppar till är tom. Stenen vi hoppade över tas då bort från planen. Målet är att komma så långt ovanför linjen som möjligt.



Uppvärmning (10 min)

1. Hur kan vi göra för att komma

   (a) ett steg ovanför linjen?

   (b) två steg ovanför linjen?

   (c) tre steg ovanför linjen?

   (d) fyra steg ovanför linjen?

Rad 5

2. Vi ska nu visa att det är avsevärt mycket svårare att nå rad 5. I den här uppgiften får du visa detta, steg för steg. I slutet går vi igenom det fullständiga beviset tillsammans. För att göra det kommer vi betrakta en målruta som ligger på femte raden ovanför linjen, och ge den vikten $x^0 = 1$. En ruta som är n steg ifrån den får vikten x^n.

   (a) Visa att i ett hopp så kommer den totala vikten av alla rutor med stenar i ändras med ett tal på formen $(x^2 - x - 1)x^n$, $-x^n$ eller $(1 - x - x^2)x^n$, för något n.

   (b) Bestäm vilka värden på x som gör så att den totala vikten av alla rutor med stenar i aldrig ökar. *Denna typ av invariant kallas ibland även* ***monovariant****.*

(c) Visa att om $0 < x < 1$ och $x^2 - x - 1 = 0$, så är

$$x^2 + x^3 + x^4 + \ldots = \sum_{n=2}^{\infty} x^n = 1$$

(d) Välj ett lämpligt värde för x från deluppgift (b) och bestäm vad summan blir av alla tal på raden precis under målraden, alltså den fjärde raden ovanför linjen.

(e) Vad blir summan av alla tal på raden som är k steg under målraden?

(f) Vad är summan av alla tal under målraden?

(g) Visa att vi aldrig kan nå femte raden med ett ändligt antal stenar.

(h) Börja nu med att fylla alla rutor under linjen med (oändligt många) stenar. Visa att resonemanget inte nödvändigtvis håller längre. Kan vi lägga till en begränsning så att vårt resonemang håller igen?

(i) Gå in på `https://www.chiark.greenend.org.uk/~sgtatham/solarmy/`

4. Cardinality I

28 juni

Motivational Question Does there exist a set of natural numbers, such that a computer program can not output exactly this list?

1. Låt oss betrakta två mängder $X = a,b,c,d,e,f,g,h$ och $Y = A,B,C,D,E,F,G,H$. Valentina påstår att mängderna är lika stora (dvs innehåller lika många element), men kan inte förklara varför för Johan eftersom han inte kan räkna med så stora tal. Kan du förklara varför mängderna har lika många element utan att räkna dem?

2. Kan du ge en definition på när två mängder har lika många element, (utanatt involvera att verkligen räkna elementen)?

3. Testa din definition!

1. Visa att varje mängd är lika stor som sig själv.
2. Visa att om mängden A är lika stor som mängden B och om mängden B är lika stor som mängden C så är mängden A lika stor som C.

Vi säger alltså att två mängder X,Y är lika stora om vi kan hitta en funktion f från X till Y så att $f(x_1) = f(x_2)$ innebär att $x_1 = x_2$ (två olika saker måste hamna på olika ställen) samt att för varje $y \in Y$ så existerar det ett $x \in X$ så att $f(x) = y$ (dvs vi träffar varje grej i y). Mindre formellt så innebär det att vi kan para ihop elementen från X och Y så att alla får precis en "kompis".

4. Visa att följande par av mängder är lika stora:

1. Positiva heltal och heltal.
2. Reella tal x sådana att $0 < x < 1$, och reella tal y sådana att $1 < y < 10$.
3. Positiva reella tal x och reella tal y sådana att $0 < y < 1$.
4. Reella tal och reella tal utom nollan.
5. Heltal och rationella tal (numbers that can be written as a fraction).
6. Reella tal och par av reella tal (dvs är planet lika stort som linjen)?

5. En talföljd är en sekvens av tal (kan vara heltal, reella tal etc beroende på situationen, i denna uppgift så antar vi att det är heltal). Ett exempel kan vara följden 1, 2, 3, 4,... Visa att samtliga talföljdsmängder nedan är uppräkneliga.

1. Mängden av talföljder som efter ett tag blir 0.
2. Mängden av talföljder som efter ett tag blir konstant (exempelvis 1, 2, 3, 3, 3, 3, ...).

3. Mängden av talföljder som efter ett tag blir periodiskt (dvs den börjar upprepa sig, t.ex. 1, 2, 3, 4, 3, 4, 3, 4,...).

6. Visa att mängden av alla talföljder (av heltal) INTE är lika stor som mängden av alla heltal.

7. Visa att det inte finns lika många heltal som reella tal.

8. Let A be a set of circles in the plane, so that no two circles intersect each other. What can be the size (cardinality) of A?

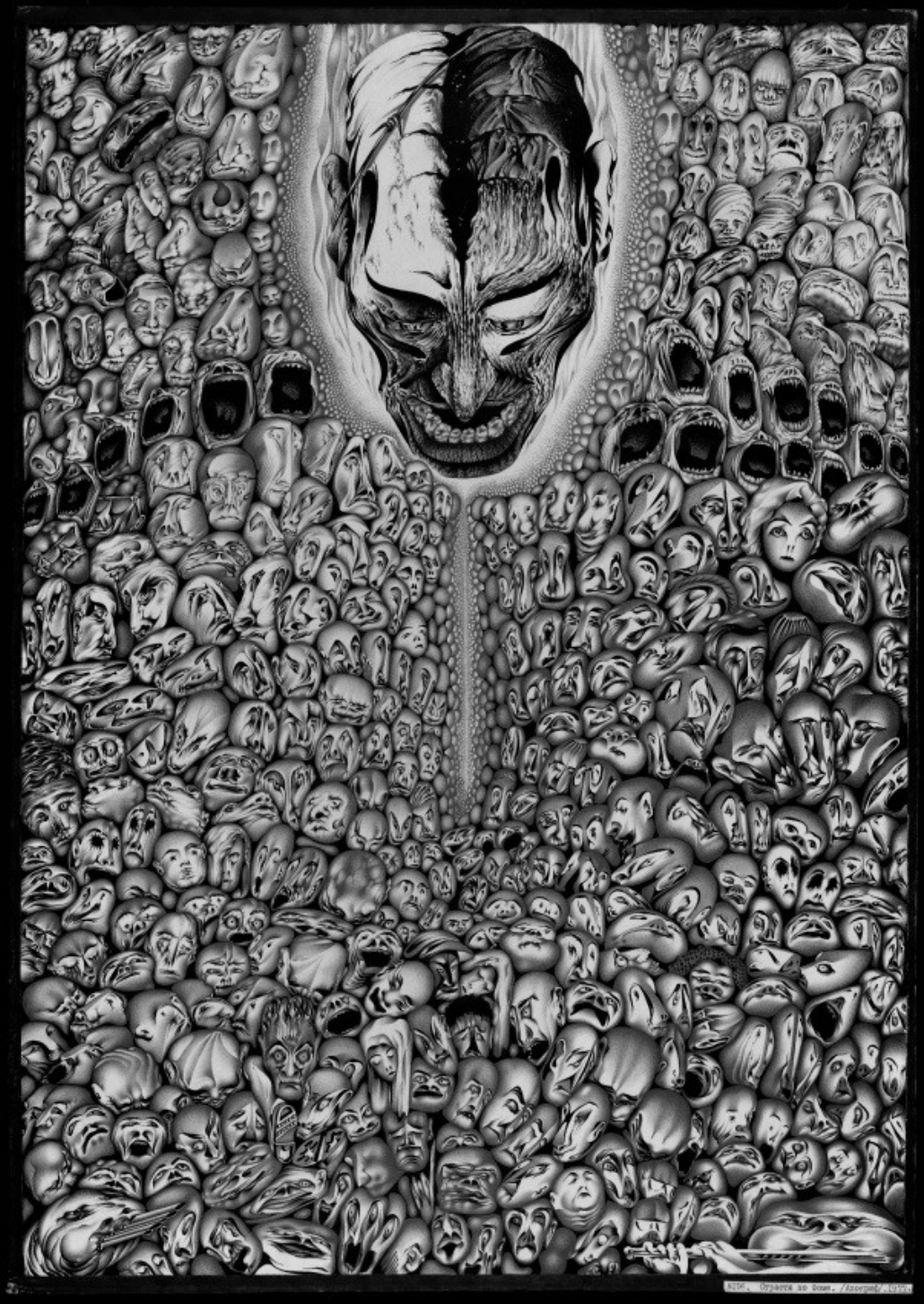
9. Let A be a set, and let $P(A)$ be the set of subsets of A. Can A and $P(A)$ have the same cardinality?

10. Can you come up with a definition for when a set A is has smaller cardinality than a set B?

11. Testa din definition!

1. If $|A| < |B|$ and $|B| < |C|$ then $|A| < |C|$
2. If $|A| < |B|$ and $|B| < |A|$ then $|A| = |B|$

($|A|$ denotes the cardinality of A).



Mathematical Infinity
by Antoly Fomenko

5. Cardinality II

29 juni

Definition. Two sets A, B have the same cardinality, if there exists a bijective map from $|A|$ to $|B|$. We write $|A| = |B|$.

1. Prove that the following definitions of $|A| \leq |B|$ are equivalent:
   - There exists an injective map from $|A|$ to $|B|$.
   - There exists a surjective map from $|B|$ to $|A|$.
2. Prove that if $|A_1| \leq |A_2|$ and $|A_2| \leq |A_3|$ then $|A_1| \leq |A_3|$.
3. Prove that if $|A| \leq |B|$ and $|B| \leq |A|$ then $|A| = |B|$.
4. Using the theorem above, is it easier to prove some of the problems from yesterday? For instance:
   - Reella tal och reella tal utom nollan.
   - Let A be a set of circles in the plane, so that no two circles intersect each other. What can be the cardinality of A?
5. Prove that for all sets A, we have $|A| < |P(A)|$.

Definition. A subset A of $\mathbb{N}$ is enumerable if there exists a computer program that lists all of the elements of A (in any order). The list might be infinite, in which case the computer program will run indefinitely.

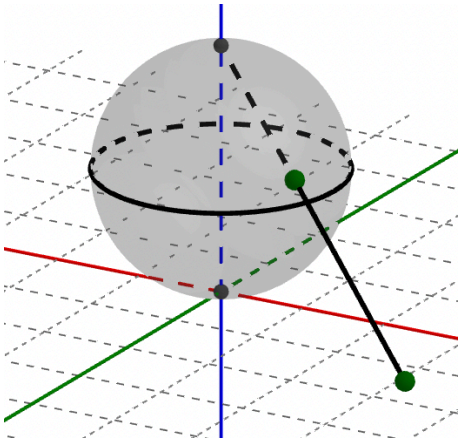
Definition. A subset A of $\mathbb{N}$ is computable if there exists a program that takes an integer x as input and returns 1 if $x \in A$, and 0 if $x \notin A$.

6. Is every computable set enumerable?
7. Let A be an enumerable set, such that the compliment of A is also enumerable. Is A necessarily computable?
8.
   a) Let A be an enumerable set. Prove that there exists a program, taking a positive integer x input, such that the program stops running if and only if $x \in A$.
   b) If such a program exist, is A necessarily enumerable?

6. Riemann Circle

30 juni

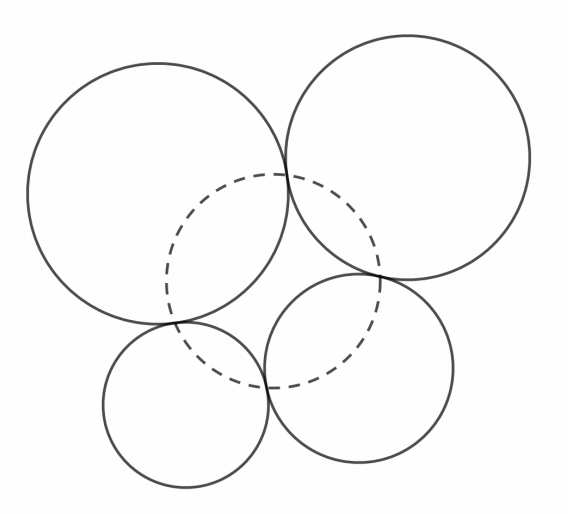
The Riemann sphere maps the plane to a sphere as shown below.

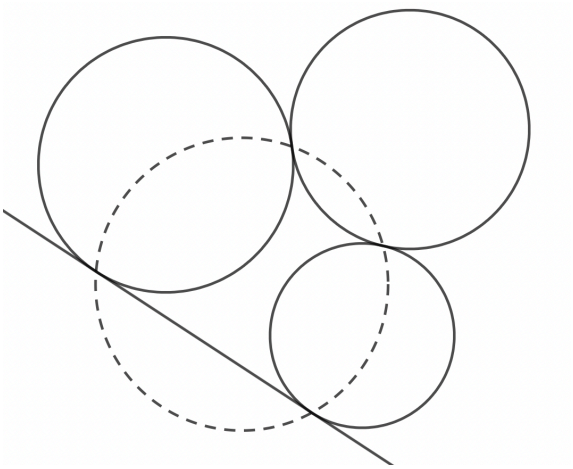


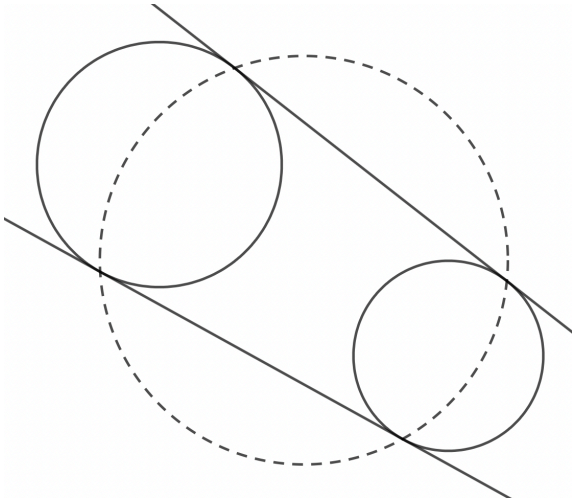
We can map the plane to the sphere, rotate the plane, and then map the sphere back to the plane. This is a transformation of the plane, called inversion.

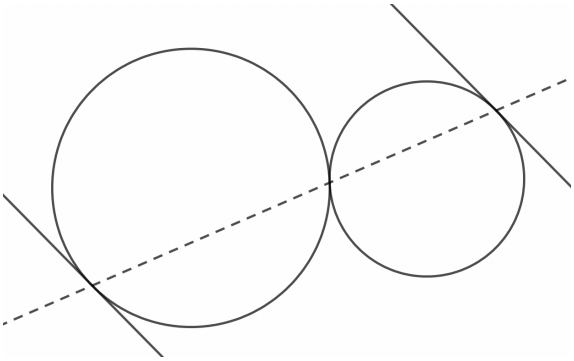
1. The transformation is almost one-to-one. But one point disappears (is not sent to any other point), and another point appears (it is not the image of any other point). Which points? We call the point that disappears for X.
2. Let ℓ be a line through X. What is the image of ℓ after the transformation?
3. Let Ω be a circle through X. What is the image of Ω after the transformation?
4. (*) What is the image of an arbitrary circle of a line, under the transformation? What happens to angels under the transformation?

5. The following four problems look very similiar, maybe you can see that their proves are basically the same? How can they be sent into each other using the transformation above?

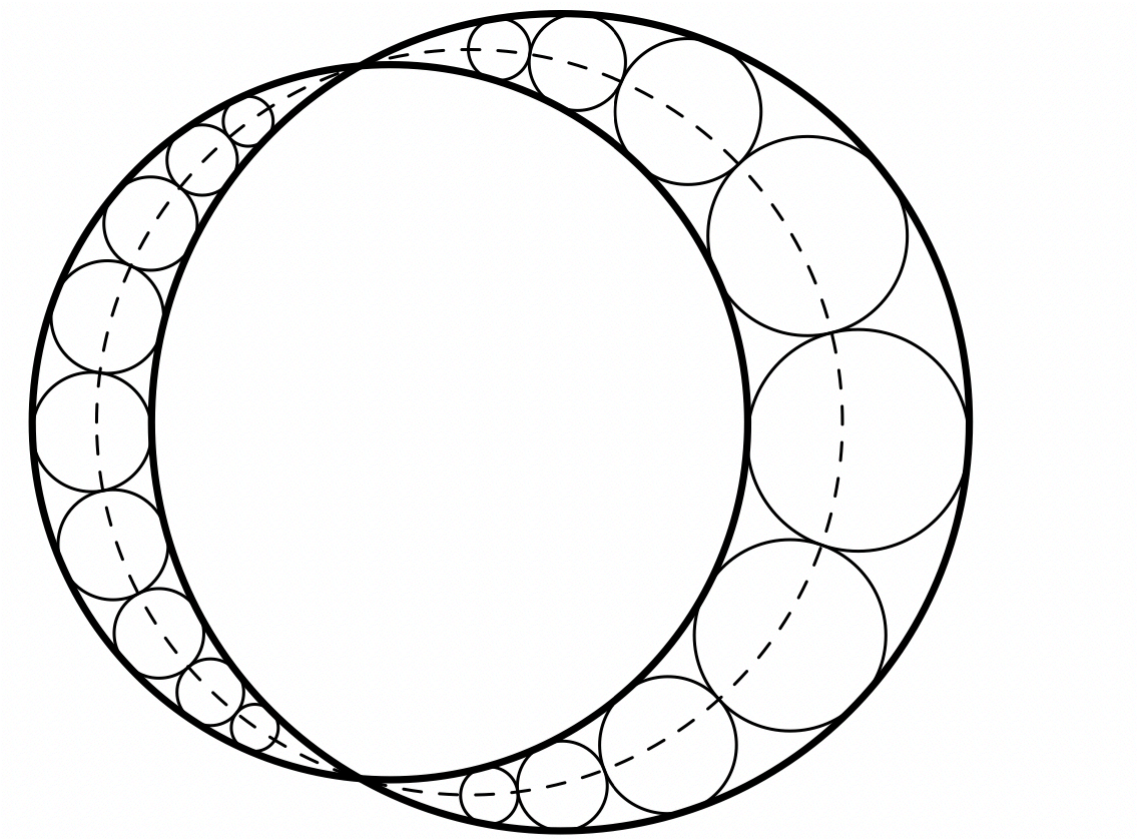








6.



7. (NMC 2007 Upgave 4) En linje genom en punkt A skär en cirkel i två punkter B och C på sådant sätt att B ligger mellan A och C. Från punkten A dras de två tangenterna till cirkeln. De träffar cirkeln i punkterna S och T. Låt P vara skärningen mellan linjerna ST och AC. Visa att $AP/PC = 2 \cdot AB/BC$.

8. Look through Akopyan's geometry in figures https://users.mccme.ru/akopyan/papers/EnG
What problems can you solve using this technique?

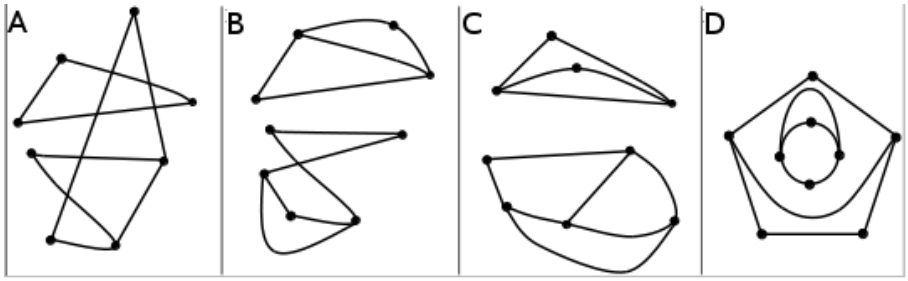
9. How can this theory be generalized to higher dimensions?

7. Grafteori 1: Rita en graf

2 juli

Uppgifter för gruppdiskussion

1. I vårt solsystem finns 8 planeter och Pluto. Följande rutter erbjuds av rymdturismen (både fram och tillbaka): Jorden-Merkurius, Pluto-Venus, Jorden-Pluto, Pluto-Merkurius, Merkurius-Venus, Uranus-Neptunus, Neptunus-Saturnus, Saturnus-Jupiter, Jupiter-Mars, Jupiter-Neptunus, Mars-Uranus. (a) Går att ta sig från Jorden till Mars på något sätt? (b) Vilka av scheman nedan kan representera ruttsystemet?



(c) Finns någon planet på schemat som man kan bestämma namnet på?

2. I en klass finns 30 elever, var och en har exakt två vänner i klassen. Vilket är det maximala antalet par man garanterat kan organisera en dans med om de två personerna i varje par måste vara kompisar?

3. (a) På ett schackbräde finns två likadana pjäser. På ett drag får man flytta en av pjäserna till en angränsande ruta vågrätt eller lodrätt. Kan pjäserna stå spegelsymmetriskt kring mittlinjen om man jämför med startpositionen efter exakt 101 drag? (b) Samma fråga, men för fem likadana pjäser.

4. Benjamin ritar upp en 6×6-kvadrat och målar sedan dess rutor, en i taget. Varje gång han målar en ruta så skriver han in ett tal i den rutan som är lika med antalet redan målade grannrutor till den. När Benjamin är färdig med att måla hela kvadraten så summerar han alla talen i rutorna. Visa att Benjamin kommer få samma summa oavsett i vilken ordning han målar rutorna.

Uppgifter att lösa enskilt

5. I en klass finns några elever, bland andra Klara och Felix. När en av eleverna fyller år (alla fyller år olika dagar) så bjuder hen in alla sina bekanta i klassen. På födelsedagskalaset lär alla tidigare obekanta känna varandra (och det är det enda sättet att lära känna varandra). Man vet att under ett år så lärde inte Klara och Felix känna varandra. Visa att de inte kommer lära känna varandra i framtiden heller.

6. Det finns n stenar som alla väger olika. På en vägning på en balansvåg så kan man jämföra vikten på två stenar. Vilket är det minsta antalet vägningar som behövs för att garanterat hitta den lättaste stenen?

7. Det finns tio tal $a_1, a_2, \ldots, a_{10}$. Man vet att det finns minst 37 heltal bland de parvisa summorna $a_i + a_j$ $(i \neq j)$. Visa att alla talen $2a_1, 2a_2, \ldots, 2a_{10}$ är heltal.

8. I en hög finns 25 stenar. Högen delas upp i två högar, sedan delas en av högarna upp i två och så vidare tills man får 25 enskilda stenar. För varje delning skriver man upp produkten av antalet stenar i de två nya högarna som man fick. Bestäm summan av alla tal man skrivit upp när man är klar med alla delningar.

9. (a) Två likadana pjäser står på ett schackbräde. På ett drag får man flytta en av pjäserna en ruta lodrätt eller vågrätt. Kan pjäserna komma till en position som är symmetrisk till den ursprungliga relativt mittlinjen på exakt 2023 drag?

(b) Fem likadana pjäser står på ett schackbräde. På ett drag får man flytta en av pjäserna en ruta lodrätt eller vågrätt. Kan man få en position som är rotationssymmetrisk med den ursprungliga på exakt 2023 drag?

10. Karin har fått tag på en trasig passare där man inte kan ändra avståndet mellan skänklarna. Hon lyckades ställa passaren på ett rutnät så att båda spetsarna hamnade i punkter med heltalskoordinater. Sedan bestämde hon sig för att flytta passaren på så sätt att ena spetsen satt kvar medan den andra flyttades till en ny punkt med heltalskoordinater. Sedan växlar hon vilken spets som står still i nästa drag medan en andra byter heltalspunkt och så vidare. Kan hon genom sådana drag uppnå en situation är passaren står där den fanns från början fast spetsarna har bytt plats?

8. Grafteori 2: Se vägar och cykler

3 juli

Bra-att-ha-fakta-1: Om graden för varje nod i en ändlig graf är lika med 2, så kan den delas upp i cykler som varken har gemensamma noder eller kanter.

Bra-att-ha-fakta-2: Om graden för varje nod i en ändlig graf är högst 2, så kan den delas upp i cykler och kedjor som varken har gemensamma noder eller kanter.

Uppgifter för gruppdiskussion

1. 20 elever löste 20 problem. Man vet att varje elev löste exakt 2 problem och att varje problem löstes av exakt 2 elever. Visa att man kan få eleverna att redovisa var sitt problem på tavlan så att var och en redovisar ett problem hen löste.

2. På en fotbollsturnering ska alla lag spela mot alla andra en gång. Efter några dagar med matcher visade det sig att man kunde ordna vilka fem lag man än tog i en cirkel så att varje lag redan hade spelat mot både laget till vänster om dem och till höger om dem. Visa att man kan spela klart fotbollsturneringen på 3 dagar. (Varje lag kan spela högst en match per dag).

Uppgifter att lösa enskilt

3. Det finns 20 pärlor av 10 färger, två pärlor av varje färg. De lades på något sätt i 10 tändsticksaskar, med 2 pärlor i varje ask.

(a)Visa att man kan välja en pärla från varje ask så att man väljer exakt en pärla av varje färg.

(b) Visa att antalet sätt att välja pärlorna så som i punkt (a) är en nollskild tvåpotens.

4. I ett gäng med barn har varje barn minst d vänner. Visa att man kan bilda
(a) en kedja (b) en ring
med åtminstone $d+1$ barn om bara barn som är vänner får hålla hand.

5. I ett land finns det enkelriktade vägar mellan vissa städer. Varje stad har lika många vägar som leder in som leder ut ur den. Man kan ta sig från vilken stad som helst till vilken annan stad som helst, kanske genom att bryta mot trafiklagarna ibland. Visa att man kan även göra det utan att bryta mot trafiklagarna.

6. I en kinesisk telefonkatalog har varje telefonnummer längden n och består bara utav nollor och ettor. Var och en av 2^n kineser ringde till alla nummer som skiljde sig med endast en siffra från deras egna. Visa att alla kineser kan bilda en ring så att de som står bredvid varandra redan har pratat.

7. I en sammanhängande graf G finns en nod med graden 101. Visa att alla kanterna kan färgas i svart och vitt så att för varje nod så skiljer sig det vita gradantalet och det svarta med som mest 1.

9. Grafteori 3: Beräkningar på grafer

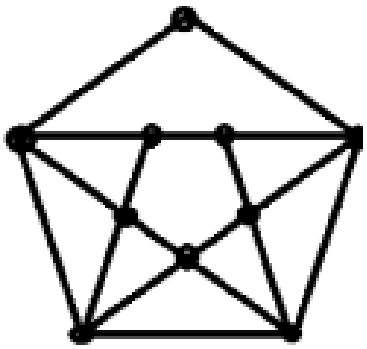
4 juli

Notation. I en graf låt E stå för antalet kanter (*edges*), V för antalet noder (*vertices*) och C för antalet sammanhängande komponenter (*components*).

Bra-att-ha-fakta 1. (a) I ett träd är $E = V - 1$. (b) I en graf utan cykler är $E = V - C$.

Bra-att-ha-fakta 2. (a) I en sammanhängade graf gäller $E \geqslant V - 1$.
(b) I alla grafer gäller $E \geqslant V - C$.

1. (a) Kan kubens kanter målas i två färger så att det mellan varje par av noder finns en väg av ena men också en väg av den andra färgen?
(b) Kan kanterna i grafen på bilden färgas i två färger så att det mellan varje par av noder finns vägar av båda färgerna?



2. (a) Ett schackbräde byggdes ihop av tändstickor, där en tändsticka är en sida på en ruta. En skalbagge sätts i en av rutorna. Den kan inte krypa över en tändsticka. Hur många tändstickor som minst måste man ta bort för att skalbaggen ska kunna krypa till alla schackbrädets rutor?
(b) En oljeshejk delade upp sitt envåningspalats i 64 likadana kvadratiska rum och delade sedan upp rummen i sju lägenheter (genom att sätta in dörrar i vissa väggar mellan rummen). I varje lägenhet bosatte sig en fru till shejken. Varje fru kan gå till alla rummen i sin lägenhet utan att gå in till de andra fruarna. Vilket är det minsta antalet dörrar som shejken behövde sätta in?

3. Det finns n stenar som alla väger olika. På en vägning på en balansvåg får man jämföra två av stenarna. En expert vet hur mycket varje sten väger, men domstolen vet inte det. Genom vägningar på balansvågen måste experten bevisa att den stenen han har pekat ut är den lättaste. Vilket är det minsta antalet vägningar experten behöver?

4. I landet Borginia ligger alla städer på en järnvägsring. Dessutom så finns det extra järnvägar från huvudstaden till alla de andra städerna, förutom dem som den redan är kopplad med längs med ringen. Borginias borgmästare delade upp järnvägsnätetet i spår mellan två anslutna städer. Dessa spår skulle säljas till två konkurrerande bolag, men på så sätt att man skulle kunna ta sig mellan valfria två städer både bara via ena bolagets spår och bara via andra bolagets spår. Är det möjligt att sälja spåren på det här sättet?

5. En klass har 30 elever. På en månad presenterades det 29 mattelösningar muntligt och varje gång var det ett par av elever framme vid tavlan. Visa att man kan dela ut poäng från 1 till 5 till varje elev så att åtminstone en får en femma och dessutom kommer summan i varje redovisningspar vara 8.

6. Mormor bakade en paj till släktträffen. Hon vet inte exakt hur många som kommer på släktträffen, men hon vet att det antingen blir 17 eller 20. Vilket är det minsta antalet bitar (inte nödvändigtvis lika stora) ska hon dela upp pajen i, för att det ska kunna delas lika mellan alla släktingar oavsett?

10. Dynamiska system I

1 juli

Definition. Ett *dynamiskt system* är ett system som förändras över tid.

Dynamiska system används för att modellera exempelvis biologiska populationer, nationalekonomi eller fysikaliska fenomen. Det är en väldigt generell formulering av en observation, och vi kommer under dessa lektioner att studera olika verktyg som hjälper oss förstå intressanta saker om dessa system.

Definition. En *differentialekvation* är en ekvation som relaterar okända funktioner och dess derivator.
Exempel:

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = 0, \quad x = x(t)$$

Eftersom vi kommer att skriva många tidsderivator använder vi den kortare notationen $\frac{dx}{dt} = \dot{x}$ (och $\frac{d^2x}{dt^2} = \ddot{x}$ osv.).
En derivata beskriver hur en funktion förändras.

Generellt sett är det enklare att hantera första ordningens termer, även om det innebär fler ekvationer. Vi kommer därför ofta göra omskrivningar på formen:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, \ldots, x_n) \\ \dot{x}_2 &= f_2(x_1, \ldots, x_n) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, \ldots, x_n) \end{aligned} \tag{10.1}$$

Exempel: Låt $\dot{x} = f(x) = 1 - x^2$. Hitta systemets fixpunkter och klassificera dem.
Lösning: Skissa $\dot{x}$ mot x.

1. Klassificera alla fixpunkter till de dynamiska systemen nedan genom att skissa fasrummet.
Tips: *Det kan vara lärorikt att skissa $\dot{x}$ mot x för hand, men om du vill ha hjälp kan du gå in på:*
`www.desmos.com/calculator`

(a) $\dot{x} = 5x + 5$

(b) $1 + x + x^2 + \dot{x} = 0$

(c) $\dot{x} = x(1 - x^2)$

(d) $5\dot{x} + 2x^{100} - 2 = 0$ *Fortsättning på nästa sida.*

(e) $\dot{x} = x^2$

(f) $\dot{x} = x^3$

(g) $\dot{x} = (x-\pi)(x-2)(x-1)(x)(x+1)(x+2)(x+\pi)$

2. Konstruera ett dynamiskt system sådant att:

(a) Varje reelt tal är en fixpunkt

(b) Varje heltal är en fixpunkt

(c) Inga fixpunkter finns

(d) Det finns exakt 3 fixpunkter, som alla är stabila

(e) Det finns exakt 100 fixpunkter.

3. Studera fixpunkterna för $\dot{x} = e^x - \cos x$. *Det räcker att du hittar deras ungefärliga position.*

4. **Fallskärmshopp:** En fallskärmshoppare accelererar mot jorden genom tyngdaccelerationen men upplever samtidigt luftmotstånd proportionerligt mot sin hastighet i kvadra, enligt

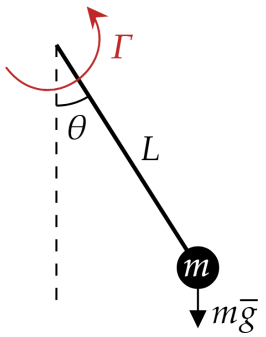
$$m\ddot{x} = mg - k\dot{x}^2, \quad k = \frac{1}{2}C\rho A$$

Hitta systemets fixpunkter och förklara hur falskärmshopparens rörelse beskrivs av ekvationen.

5. **Överdämpad pendel:** Utslagsvinkeln θ av en pendel under inverkan av en yttre kraft Γ (vridmoment) kan fysikaliskt skrivas som

$$mL^2\ddot{\theta} + B\dot{\theta} + mgL\sin\theta = \Gamma$$

där B är en dämpningsfaktor. För en överdämpad pendel gäller $\ddot{\theta} \approx 0$. Hur kommer systemet bete sig? Har systemet några fixpunkter?



11. Dynamiska system II: Bifurkationer

3 juli

Bifurkationer

Definition. Ett envariat dynamiskt system som har en extern parameter har en bifurkation (latin: *bifurcus* - tvågrenad) om antalet fixpunkter/jämviktslägen förändras när den externa parametern ändras.

Exempel: fri parameter r:

$$\begin{aligned} &\dot{x} = r + x^2 \\ &r < 0 : 2 \text{ fixpunkter} \\ &r = 0 : 1 \text{ fixpunkter} \\ &r > 0 : 0 \text{ fixpunkter} \end{aligned} \tag{11.1}$$

1. Skissa ett **bifurkationsdiagram**, det vill säga fixpunkternas position och typ ritat i x-r-planet, för exemplet ovan.

2. Skissa bifurkationsdiagrammet för $\dot{x} = rx - x^2$ (r är den externa parametern).

3. Konstruera ett dynamiskt system som har en fixpunkt som delar sig i *tre* när en parameter ändras, och skissa dess bifurkationsdiagram.

4. Skissa funktionen och rita bifurkationsdiagrammet för $\dot{x} = 1 + rx + x^2$.
Ledning: Dela upp funktionen i två delar.

5. **Frivillig:** Hitta på en exotisk funktion med spännande bifurkationsdiagram.
Exempel: $ke^{x/k} - \cos x$, $e^{xk} - k\cos(x)$

Ett exotiskt system: diskreta steg

6. Låt oss nu studera ett system som är av mer spännande matematisk karaktär:

$$f(x) = 10 \cdot x - \lfloor 10 \cdot x \rfloor \quad \lfloor \cdot \rfloor \text{ är den s.k. ”floor-funktionen”: avrnundning nedåt.} \tag{11.2}$$

Till exempel: $f(0.1234) = 1.234 - 1 = 0.234$.

(a) Rita fasrummet $f(x)$ mot x.

(b) Hitta alla fixpunkter $f(x) = x$.

(c) Vi säger att f har en cykel av längd k om $f^{(k)}(x) = f(f(\ldots_{[\text{k gånger}]} f(x)\ldots) = x$
Visa att f har cykler av alla längder.

(d) Visa att f har oändligt många aperiodiska vägar, dvs $f^{(k)}(x) \neq x$ för alla k.

12. Fraktaler, dimensioner & oändligheter

4 juli

Dagens pass är ett 1.5x *superpass!* Vi ska prata om fraktaler, oändligheter och dimensionsbegreppet. Vi kommer idag inte se kopplingarna till dynamiska system, men de finns, och framför allt är det ett väldigt kul ämne! Vi kommer vandra in på området ***mängdlära*** och ytligt diskutera begrepp som ***mått*** och ***uppräknelighet***. Allt detta knyter vi sedan ihop genom att studera ett av matematikens vackraste under: fraktaler.

Dimensioner

Innan vi börjar ifrågasätta vår verklighet värmer vi upp genom att bekanta oss med den.

1. Oliver hade en stor bit tvål i form av ett rätblock. Efter att han hade duschat 7 gånger blev biten hälften så lång, hälften så bred samt hälften så hög som den var i början. För hur många duschar till räcker den tvålbiten som är kvar?

2. En kvadrat med sidan 1 m delades upp i småkvadrater med sidan 1 cm. Hur många små kvadrater fick man?

3. En kub med sidan 1 m delades upp i småkuber med sidan 1 cm. Hur många små kuber fick man?

4. Ett klosstorn består av 3 klossar staplade på varandra. Hur mycket mer väger klosstornet om alla klossar skalas upp med en faktor 2 (bredd, höjd och djup görs dubbelt så långa)?

5. En elefantunge växer på ett år så att alla dess mått tripplas. Hur många gånger mer väger elefanten efter ett år? Hur många gånger mer hud har elefanten?

6. Vad väger mer: ett järnklot med radie 8, eller två järnklot med radie 4?

7. I problemet ovan: Vilket har störst totala area? Störst totala omkrets?

8. Mini-Valentina är $\frac{1}{2}$ meter lång och har knutit ett snöre runt jorden, men nu snubblar hon på det hela tiden. Hur mycket längre behöver snöret vara för att istället hänga 1 meter över jordens yta överallt?

Oändligheter (kardinalitet)

Vi ska kort diskutera konceptet **uppräknelighet**. Ni kan ha gått igenom detta med Alex, i så fall blir detta en snabbrepetition.

Definition. *(Slarvig) En mängd sägs vara* ***uppräknelig*** *om man kan räkna upp dess element. Mängder kan även vara* ***oupräkneliga****.*

Ett enkelt exempel vore en ändlig mängd: $\{\pi\}$ är trivialt uppräknelig: *"π är*

det första elementet. Klar!". En annan typiskt uppräknelig mängd är de positiva heltalen $\{1,2,3,...\} = \mathbb{Z}^+$: *"1 är det första elementet, 2 det 2:a, 3 det 3:e, och så vidare."* Vi behöver alltså inte kunna "räkna klart", bara vi kan räkna och, om vi skulle räkna för evigt, räkna in alla element i mängden. Vissa av er har kanske hört talas om *Hilberts hotell*, som täcker denna idé!

* = *extraproblem (kluriga!)*

9. Visa att mängden $\{\pi\} \cup \mathbb{Z}^+ = \{\pi, 1, 2, 3, ...\}$ är uppräknelig.

10. Visa att även de naturliga talen $\mathbb{N} = \{..., -2, -1, 0, 1, 2, ...\}$ är uppräkneliga genom att hitta en metod att räkna dem på ett ordnat sätt.

11. Visa att mängden av alla heltalskoordinater (a, b), där $a, b \in \mathbb{N}$ är uppräknelig. *Ett annat sätt att säga detta är: Visa att* $|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}| = |\mathbb{N}^2|$

12. Visa att de rationella talen $\mathbb{Q} = \{\frac{p}{q} \quad | \quad p, q \in \mathbb{Z} \text{ och } q \neq 0\}$ (bråktalen) är uppräkneliga.

13. * Visa att mängden av alla tal på formen $2^a 3^b 5^c$, $a, b, c \in \mathbb{Z}$ är uppräknelig.

14. * En groda hoppar på heltalslinjen. Den börjar på något okänt heltal, och hoppar med en okänd men konstant (heltals-)längd i någon fix riktning. Du får i varje drag titta på ett tal och om grodan är där vinner du. Utforma en strategi som gör att du garanterat kan hitta grodan på ett ändligt antal drag.

15. * Visa att $|\mathbb{N}| = |\mathbb{N}^n| \quad \forall n \in \mathbb{Z}^+$

Man brukar säga att dessa uppräkneliga men oändliga mängder som vi betraktat ovan utgör den "minsta" oändligheten, och vi kallar den för $\aleph_0$ (uttalas *aleph-noll*). Man skriver ibland $|\mathbb{N}|$ för att beteckna en mängds "storlek". Om en mängd M har $|M| = |\mathbb{N}|$ så är M uppräknelig. I någon bemärkelse finns det alltså lika många positiva heltal som det finns bråktal! *Vi har i uppgifterna ovan visat att* $\aleph_0 + 1 = \aleph_0$, *att* $2 \cdot \aleph_0 = \aleph_0$ *och till och med att* $\aleph_0 \cdot \aleph_0 = \aleph_0$. *Wow!*

En ***oupräknelig*** mängd däremot är strikt större än en uppräknelig. Ett exempel på en oupräknelig mängd är de rationella talen $\mathbb{R}$. Det vill säga, det finns *strikt fler* reella tal än heltal: $|\mathbb{R}| > |\mathbb{N}|$. Kontinuumhypotesen förutspår att *kardinaliteten* ("storleken") av de reella talen är $\aleph_1$ (det vill säga, det finns ingen oändlighet med storlek mellan $|\mathbb{N}|$ och $\mathbb{R}$). Man kan t.ex. även visa att $2^{\mathbb{N}}$, $\mathbb{R}$ och $\mathbb{C}$ alla har samma kardinalitet!
Jag visar ett bevis för att $\mathbb{R}$ är oupräknelig på tavlan. *Om du vill försöka själv är bliden till höger en ledning.*

$$
\begin{array}{|l|}
\hline
s_1 = 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\ldots \\
s_2 = 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\ldots \\
s_3 = 0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\ldots \\
s_4 = 1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\ldots \\
s_5 = 1\,1\,0\,1\,0\,1\,1\,0\,1\,0\,1\,1\ldots \\
s_6 = 0\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0\,1\ldots \\
s_7 = 1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,1\ldots \\
s_8 = 0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\ldots \\
s_9 = 1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,1\ldots \\
s_{10} = 1\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\ldots \\
s_{11} = 1\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0\,1\ldots \\
\vdots \quad \vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\vdots\;\ddots \\
\hline
\end{array}
$$

$$
\begin{array}{|l|}
\hline
s = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,1\ldots \\
\hline
\end{array}
$$

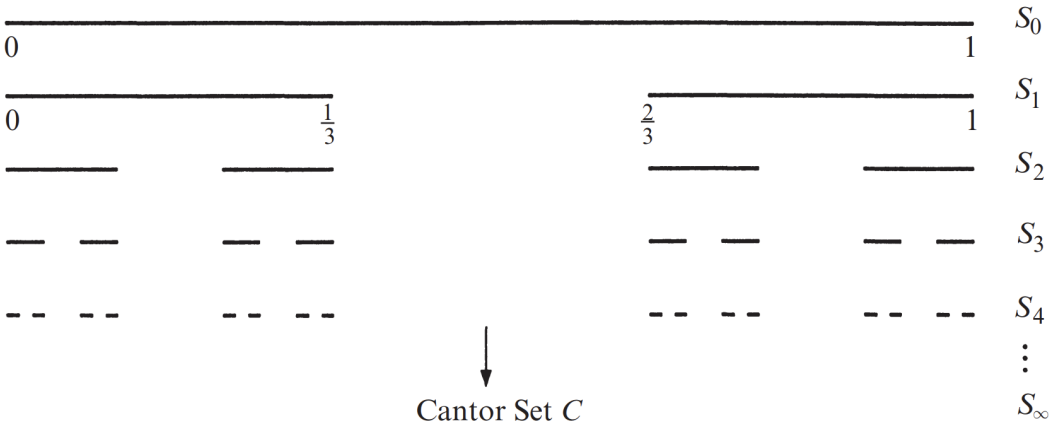
16. * Visa att $|2^{\mathbb{N}}| = |\mathbb{R}|$.

Fraktaler

Definition. En *fraktal* är en komplex geometrisk form med struktur på godtycklig små skalor. De är ofta självliknande.

Cantormängden

Ett typiskt exempel är ***Cantormängden***, som har dimension ≈ 0.63! Den definieras som följer:



Börja med det slutna intervallet $S_0 = [0,1]$ och ta bort den öppna mittersta tredjedelen, dvs ta bort $(\frac{1}{3}, \frac{2}{3})$. Detta producerar mängden med två intervall som vi kallar S_1. Nu tar vi bort den mittersta tredjedelen av dessa två intervall för att skapa S_2, och så vidare. Cantormängden $C = S_\infty$.
Vi ska nu studera några fascinerande egenskaper hos Cantormängden:

17. Visa att Cantormängden har ***mått*** noll, i bemärkelsen att den kan täckas av intervall vars totala längd är godtyckligt liten.

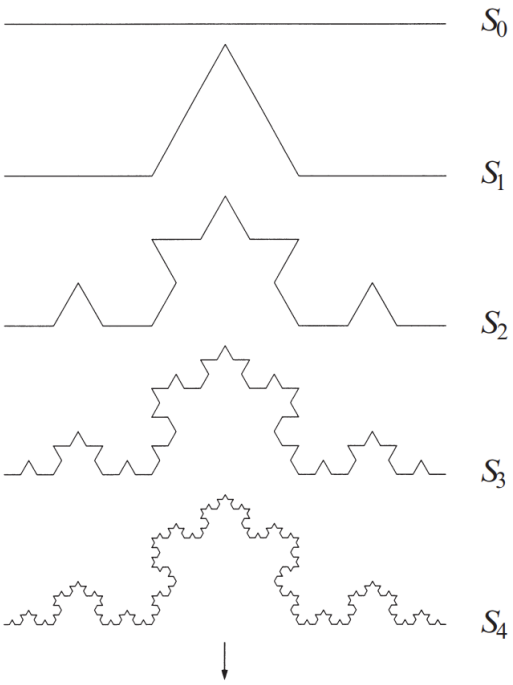
18. Visa att Cantormängden består av alla punkter $c \in [0,1]$ som inte innehåller några 1:or i sin bas-3 expansion.
Decimaltal i bas 3: $x = \frac{a_1}{3^1} + \frac{a_2}{3^2} + \frac{a_3}{3^3} + \ldots \implies x = 0.a_1a_2a_3\ldots_{tre}$ *(bas 3).*

19. I uppgift **18** sa vi att alla tal i Cantormängden kunde skrivas som ett tal i bas 3 som inte innehåller några 1:or. Vi säger dock också att $\frac{1}{3} \in C$ och $\frac{1}{3} = 0.1_{tre}$. Hur går detta ihop?

20. Med hjälp av problem **18**, visa att Cantormängden är oupräknelig.

Fraktaldimensioner

Vi tittar nu på en annan fin fraktal en stund, den så kallade ***Koch-kurvan***:



Vilken dimension tycker du vi ska säga att Koch-kurvan har? Du kanske är lockad att säga 1-dimensionell, eftersom det är en kurva, och typiskt brukar vi kunna hitta en avbildning från $\mathbb{R}^1 \to$ kurva. Men K har oändlig kurvlängd! Det är till och med så att avståndet mellan vilka två punkter som helst på K är oändligt. Så kanske är K 2-dimensionell? Det verkar också fel, eftersom vi inte ser någon uppenbar area... Alltså borde dimensionen vara mellan 1 och 2, vad det nu än det betyder?

I början av dagens lektion observerade vi att linjer skalar som r, areor som r^2, volymer som r^3 och i princip kunde vi generalisera det till att inkludera r^n för $n > 3$.

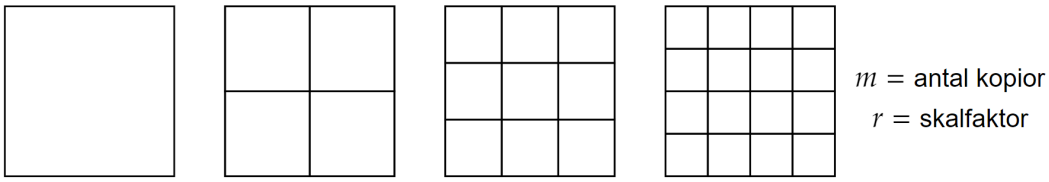
Självlikhetsdimensionen

Ett generaliserat sätt att räkna dimensioner är att betrakta ett självlikt objekt som består av m kopior av sig själv som är nedskalade med en faktor r. ***Självlikhetsdimensionen*** definieras då som:

Definition. ***Självlikhetsdimensionen*** d***:***

$$d = \frac{\ln(m)}{\ln(r)}$$

där $\ln(x)$ är den *naturliga logaritmen*. Den har egenskapen att $\ln e = 1$, och framförallt gäller att $\ln(a^b) = b\ln(a)$.



$m = 1$, $r = 1$ | $m = 4$, $r = 2$ | $m = 9$, $r = 3$ | $m = 16$, $r = 4$

Exempel: En kvadrat kan ses som a^2 mindre kvadrater, nedskalade med en faktor a. Vi får då enligt formeln ovan:

$$d = \frac{\ln(a^2)}{\ln(a)} = \frac{2\ln(a)}{\ln(a)} = 2$$

Självlikhetdimensionen verkar alltså fungera som vi väntar oss för vanliga heltalsdimensioner.

21. Visa att Kantormängden har självlikhetsdimension $\ln(2)/\ln(3)$

22. Bestäm självlikhetsdimensionen av Koch-kurvan.

23. Sierpinski har precis flyttat hemifrån och ska inreda sitt nya rum med en matta. Skräddaren han ska beställa av säger att han kan sy mattor för 1000 kr/m^2. Sierpinski, som är en matematikstudent, har såklart ont om pengar. Han planerar dock täcka upp för sina ekonomiska tillkortakommanden med hjälp av matematik. Han ber skräddaren producera en matta enligt följande instruktioner:

1. Tänk dig en 1×1 m stor matta.
2. Dela in kvadraten i 9 lika stora rutor.
3. Ta bort den mittersta rutan.
4. För de återstående 8 rutorna, upprepa steg 2-4.

Vad kostar Sierpinskis matta?

24. Konstruera en ny Cantormängd genom att ta bort mittersta halvan av varje mängd, dvs S_1 består av $[0, \frac{1}{4}]$ och $[0, \frac{3}{4}]$.

(a) Vad är självlikhetsdimensionen av denna modifierade Cantormängd?

(b) Vad är måttet?

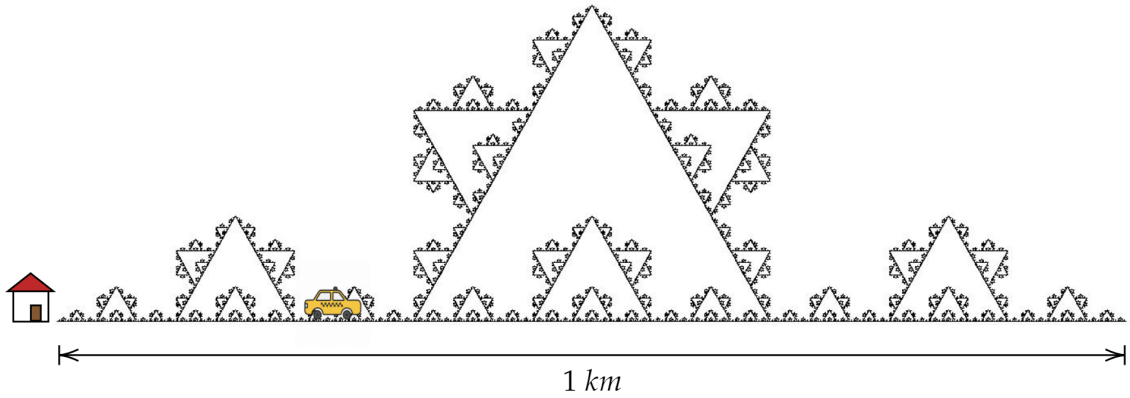
25. **Djävulens trappa:* Välj något $0 \leq x \leq 1$. Välj en slumpmässig punkt $p \in C$ (Cantormängden). Låt $P(x)$ vara sannolikheten att $p < x$.

(a) Beräkna $P(x)$ för S_0 (se definitionen av S_0 i bilden med Cantormängden ovan). Skissa sannolikheten i en graf: $P(x)$ som en funktion av x.

(b) Gör samma sak för S_1, S_2 och S_3 (låt oss kalla funktionerna $P_n(x)$ med $n = 1, 2, 3$).

(c) Hur ser $P_\infty(x)$ ut? Är den kontinuerlig? Vad har den för lutning?

26. ** Benjamin ska åka taxi på den modifierade Koch-kurvan. Kurvan ser ut som Koch-kurvan, fast basen på triangeln är ifylld (se bild nedan). Taxin kostar 10 kr/km, och den kör alltid kortaste vägen mellan två punkter. Benjamin åker varje dag från sitt hem på punkten längst till vänster på kurvan till en slumpmässig punkt på kurvan. Vad förväntas en sådan resa kosta?



13. Mattedrabbning röd

5 juli

1. På ett plan finns uppräkneligt många (punktstora) myggor. I en annan punkt i planet står djurrättsaktivisten Elin. Visa att Elin kan skjuta med en laser utan att träffa någon mygga.

2. *Kochs snöflinga:* Ta 3 kopior av Koch-kurvan, och sätt ihop dem i en triangelliknande form (S_0 är en liksidig triangel med sidan 1, S_1 en 6-uddig stjärna, osv.). Bestäm dess area.

3. Elias och Emilio försöker bygga en robot som ska kunna lösa Rubiks kub. De kan sätta ihop ett par av komponenter med en sladd om de vill. Varje par av komponenter kan sättas ihop med som mest en sladd. Så fort alla delarna blir ihopkopplade blir roboten ohanterbar och går sönder. Elias börjar med att sätta ihop två delar med sladd, Emilio kopplar med nästa sladd och så turas de om. Den som kopplar ihop en sladd som gör att roboten går sönder förlorar spelet. Vem har en vinnande strategi om roboten har 1043 delar?

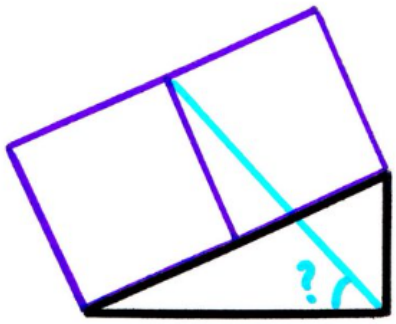
4. Den största gemensamma delaren för de positiva heltalen a, b, c är 1. Man vet att $\frac{1}{a} - \frac{1}{b} = \frac{1}{c}$. Visa att abc är ett kvadrattal.

5. Talen $a_1, a_2, \ldots, a_n$ är vardera lika med 1 eller -1. Vidare gäller

$$a_1a_2 + a_2a_3 + \ldots + a_na_1 = 0$$

Visa att n måste vara delbart med 4.

6. Två kvadrater sitter på hypotenusan till en rätvinklig triangel så som bilden visar. Hur stor är den markerade vinkeln på bilden (ange alla svar och visa att inga andra svar är möjliga)?



III

Matematik - Blå

1. Genererande funktioner I

28 juni

Ett kraftfullt verktyg för att analysera sekvenser av tal. Idén är att istället för att betrakta en sekvens av tal (t.ex. $2,3,5,8,12,\ldots,a_n,\ldots$) så betraktar vi en funktion som innehåller all information om sekvensen. Funktionen, som vi betraktar den, kommer *inte* att bete sig som $f(n)=a_n$, utan vi vill att den har egenskaper som gör att vi enkelt kan beskriva termer relativt varandra, ex. a_n, a_{n+1}, a_{n+5} etc. Ett trevligt sätt att göra detta är med hjälp av en funktion som ser ut som följer: $2+3x+5x^2+8x^3+12x^4+\ldots$ (för exemlpet ovan). Varje term i serien är alltså en koefficient för motsvarande grad av ett oändligt polynom, även kallat ***power series*** (*svenska: "potensserie"*). Vi kallar denna funktion för seriens ***genererande funktion***.

Definition. En ***genererande funktion*** till talsekvensen a_n är

$$G(a_n;x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots = \sum_{n\geq 0} a_n x^n.$$

Exponenten på potensen i indikerar direkt vilket tal i ordningen a_i är. Vidare finns det verktyg för power series som vi kan använda. Till exempel kan den genererande funktionen för sekvensen $1,1,1,\ldots$ skrivas som $1+x+x^2+\ldots=\frac{1}{1-x}$. Vi säger att sekvensen $1,1,1,\ldots$ har genererande funktion $\frac{1}{1-x}$.

1. Visa att $1+x+x^2+\ldots=\frac{1}{1-x}=G(x)$. Gäller detta $\forall x$?

2. Hitta en genererande funktion till följande sekvenser med hjälp av resultatet från uppgift **1**:

   (a) $2,2,2,2,\ldots \quad (a_n=2)$

   (b) $1,-1,1,-1,\ldots \quad (a_{n+1}=-a_n)$

   (c) $1,3,9,27,81\ldots \quad (a_n=3^n)$

   (d) $2,4,10,28,82,\ldots \quad (a_n=3^n+1)$.

3. Hitta en genererande funktion för $1,0,1,0,\ldots$.

4. Hitta en genererande funktion för $0,1,2,3,\ldots$ på två sätt, med hjälp av $G(x)$ för $1,1,1,\ldots$.

   1. Genom att nyttja derivering.

   2. Genom att nyttja summering av förskjutna sekvenser.

5. Hitta en ***operator*** (ungefär: algoritm/regel/funktion/generell metod) för att en genererande funktion för n^k för $k\geq 0$.

6. Derivera nu uttrycket från uppgift **1**. Hur kan du använda detta resultat för att bräkna det förväntade antalet gånger du behöver slå en 6-sidig tärning innan du slår en 1:a?

7. Hitta en genererande funktion för $1,1,1,1,1$ (ändlig sekvens).

8. Hitta en genererande funktion för n^2 $(1,4,9,16,...)$.

9. f är genererande funktion för sekvensen $a_0, a_1, a_2, ...$ Vad beskriver $\dfrac{f(x)}{1-x}$?

10. Hitta en genererande funktion för binomialkoefficienterna $\binom{n}{k}$. Notera att detta blir en ändlig genererande funktion, upp till ordning n. Fundera på olika sätt att komma fram till detta resultat.

11. Med hjälp av resultatet i **10**, visa identiteterna:

$$\binom{n}{0} + \binom{n}{1} + ... + \binom{n}{n} = 2^n$$

samt

$$\binom{n}{1} + \binom{n}{3} + ... = \binom{n}{0} + \binom{n}{2} + ...$$

12. Beräkna summan

$$\sum_{k=0}^{n} 2^k \binom{n}{k}$$

med hjälp av en genererande funktion. *Kan du generalisera ditt resultat?*

13. Alex har tre lådor med strumpor. En låda innehåller 2 röda strumpor, en låda innehåller 2 blåa strumpor och en innehåller 3 gröna strumpor. På hur många olika sätt kan Alex välja 4 strumpor? (Strumpor av samma färg är oskiljaktliga, ordningen på valet spelar ingen roll)

Ledning: Betrakta en genererande funktion för hur många sätt Alex kan välja n röda strumpor.

2. Genererande funktioner II

29 juni

Ett område där genererande funktioner kan vara ett kraftfullt verktyg är olika typer av kombinatorikproblem. Vi har redan introducerat detta lite i föregående lektion, då vi studerade GF för $\binom{n}{k}$.

1. Lös följande kombinatorikproblem genom att formulera om problemet i termer av genererande funktioner:
Hitta antalet lösningar till $e_1 + e_2 + e_3 = 17$ där $e_1 \in \{2,3,4,5\}$, $e_2 \in \{3,4,5,6\}$ och $e_3 \in \{4,5,6,7\}$.

2. Clara, Daniel och Elias ska dela på 8 likadana kakor. På hur många sätt kan de fördela kakorna om alla får minst två kakor, och ingen får fler än 4? *Vad blir den genererande funktionen för problemet och vilken koefficient ska studeras?*

3. Använd genererande funktioner för att bestämma antalet sätt en person kan stoppa in 1-kronor, 2-kronor och 5-kronor i en godisautomat för att betala för en vara som kostar r kronor. Studera både fallet där ordningen pengarna stoppas in inte spelar någon roll, och då det gör det. *Du behöver inte räkna fram det numeriska svaret; ställ bara upp uttrycket för den genererande funktionen och vilken koefficient du söker.*

4. Hitta en genererande funktion vars koefficient c_n är antalet delmängder av mängden $\{1,2,3,\ldots,2000\}$ vars summa är n.

5. Hitta antalet delmängder av mängden $\{1,2,3,\ldots,2000\}$ vars summa är delbar med 2 med hjälp av resultatet i **4** och verifiera ditt resultat med klassisk kombinatorik.

6. Visa att summan av *alla* lösningar, ξ_i, till ekvationen $z^n = 1$ ($n \in \mathbb{N}^+$) är noll, d.v.s. $\sum_{i=1}^{n} \xi_i = 0$. Vad gäller för $\sum_{i=1}^{n} \xi_i^2$, eller $\sum_{i=1}^{n} \xi_i^n$?

7. Hitta antalet delmängder av mängden $\{1,2,3,\ldots,2000\}$ vars summa är delbar med 5.

*Ledning: använd resultatet från uppgift **6** med $n = 5$ och betrakta sedan GF med $x, x^2, \ldots, x^5$.*

3. Genererande funktioner III

30 juni

Nu när vi har bekantat oss lite vid idén av genererande funktioner så ska vi studera ett annat område där de är användbara: att hitta slutna formler för sekvenser av tal.

1. Studera derivatorna av den genererande funktionen till $a_n = 1/n!$ Försök därefter skriva om den genererande funktionen till en mer kompakt form.

2. Hitta sekvensen för vilken $\frac{e^x + e^{-x}}{2}$ är en genererande funktion.

3. **Repetition:** f är genererande funktion för sekvensen $a_0, a_1, a_2, \ldots$. Vad beskriver $\frac{f(x)}{1-x}$? *Detta resultat kommer vara av nytta i kommande uppgift.*

4. En fiskpopulation börjar med 50 fiskar och förökar sig fyrfaldigt varje år. Samtidigt dör 100 fiskar varje år. Låt populationen år t vara p_t. Hitta en sluten formel för p_t.
Ledning: Finn den genererande funktionen för p_t och skriv om den i termer av kända genererande funktioner.

5. Betrakta sekvensen av tal $a_0, a_1, \ldots$ som uppfyller $a_{n+1} = 2a_n + 1$ med $n \geq 0$ och $a_0 = 0$. Dessa har genererande funktion $A(x)$. Hitta sekvensen med hjälp av genererande funktioner, genom att multiplicera bägge led av rekursionsrelationen med x^n och summera, och relatera detta till $A(x)$.

6. *Extra:* Gör problem **5** för sekvensen

$$a_{n+1} = 2a_n + n, \quad n \geq 0, a_0 = 1$$

7. Visa att en sluten formel för det n:te Fibonaccitalet F_n ges av:

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right) \tag{3.1}$$

Låt $F_0 = 0, F_1 = 1$ och minns att $F_n = F_{n-1} + F_{n-2}$.
Extra: Visa att även $round(\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n)$ är en sluten formel för F_n.

4. Curves and Intersections

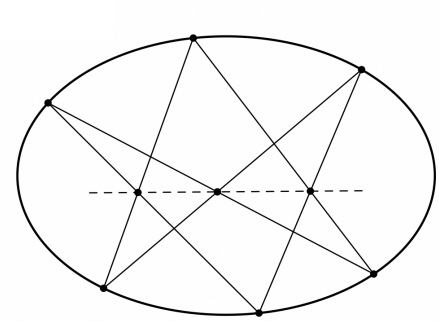
28 juni

Definition. A cubic curve is the set of points in the planethat are solutions to an equation on the form

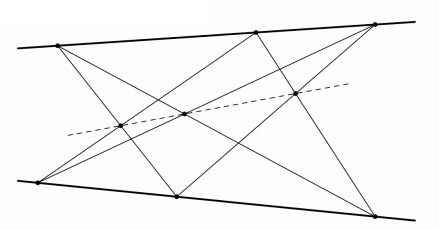
$$a_1x^3 + a_2x^2y + a_3xy^2 + a_4y^3 + a_5x^2 + a_6xy + a_7y^2 + a_8x + a_9y + a_10 = 0.$$

Cayley-Bacharach If three cubic curves intersect in 8 points, then they intersect in 9 points.

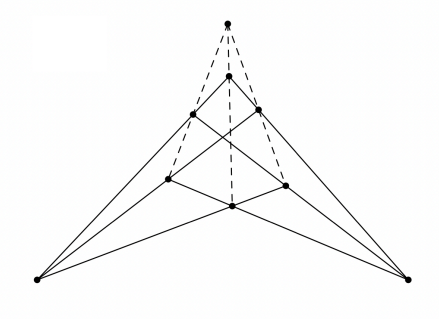
1. Pascal's theorem



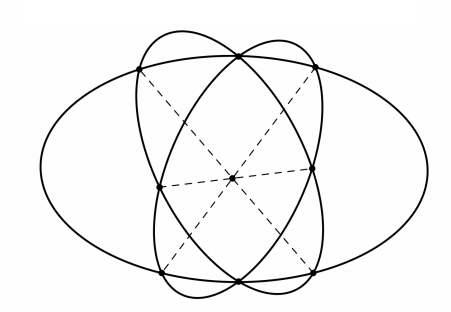
2. Pappus theorem



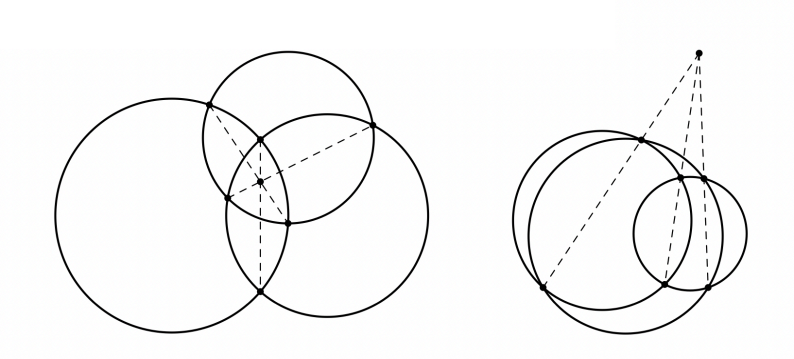
3.

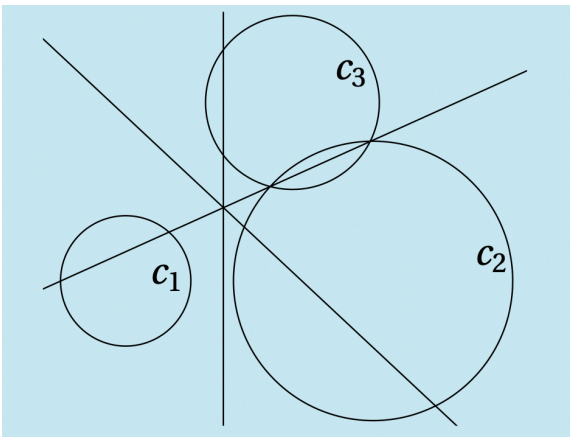


4. Three conics theorem

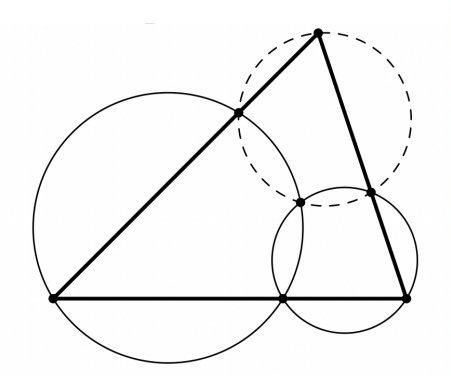


5. Radical axis theorem

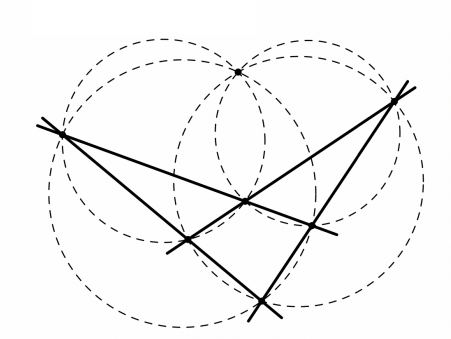




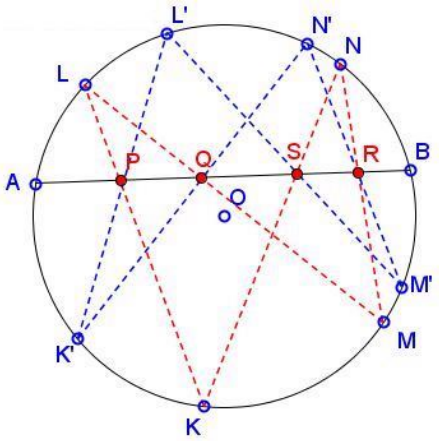
6. Miquel's theorem



7. Miquel point



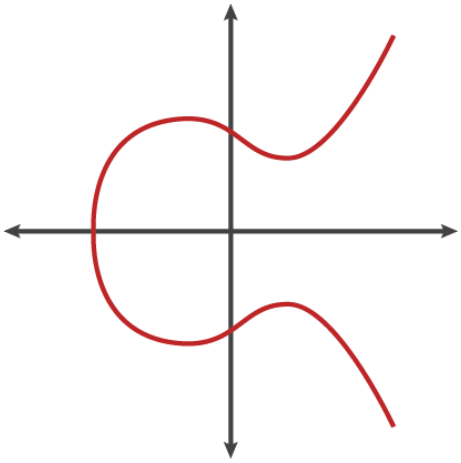
8. Two butterfly theorem



9. Let the eight vertices of an octagon lie on a conic, and alternately colour the edges red and blue. Prove that the remaining eight heterochromatic intersections of (the extensions of) the edges lie on another conic.

Definition. A group is a set G together with an operation denoted $\star$ that combines any two elements a, b in G and forms a new element $a \star b$ in G such that the following three requirements hold:
Associativity For all a, b, c in G we have $(a \star b) \star c = a \star (b \star c)$.
Identity element G contains an element e that satisfy $a \star e = e \star a = a$.
Inverse element For all a in G there exist an element b in G that satisfy $a \star b = b \star a = e$. b is called the inverse to a.

10. Let E dentoe the cubic curve $y^2 = x^3 - x + 1$. For any two points p_1, p_2 we draw the line ℓ through them. This line intersects E in a third point p, and if we reflect p across the $x-$ axis we get a new point that we define as $p_1 + p_2$. Prove that this defines a group on E. What are the inverse elements, and how do you add a point with itself? What is the zero element?



5. Homogenous Coordinates

29 juni

Bezout's Theorem Any two curve in the plane of degree d_1 and d_2 has $d_1 \cdot d_2$ intersection points when we count complex intersections, intersections at infinity, and intersections with multiplicity.

Definition. Two curves are tangent in a point, if they intersect with multiplicity greater than one.

1. Generalize homogeneous coordinates to the plane $\mathbb{CP}^2$.

2. Prove that the parabola $y = x^2$ is tangent to the line at infinity.

3. Consider the homogeneous equation a circle. Find two points that all circles goes through. These two points are called the circular points at infinity. Is a curve of degree 2, passing through these two points, necessarily a circle?

4. Consider two concentric circles. How many times should they intersect according to Bezout's theorem - and how?

5. What does the surface of complex points on a conic look like - including the ones at infinity? For instance a circle, or a hyperbola?

6. What does the surface of complex points on the cubic

$$y^2 = x^3$$

look like?

Definition. A curve is irreducible if it is not the union of two other curves.

Definition. A point on a curve is singular if any line through it intersects it with multiplicity more than one.

7. Show that an irreducible algebraic curve of degree 4 can have atmost 3 singular points.

8. What is the maximum possible number of singular points on an irreducible algebraic curve of degree d?

9. Prove and test that a curve defined by an equation in homogeneous coordinates

$$P(x_0, x_1, x_2) = 0$$

is singular in (x_0, x_1, x_2) if and only if

$$\frac{\partial P}{\partial x_0} = \frac{\partial P}{\partial x_1} = \frac{\partial P}{\partial x_2} = 0.$$

10. How can you tell, from partial derivatives, if a curve has singular points of higher multiplicity?

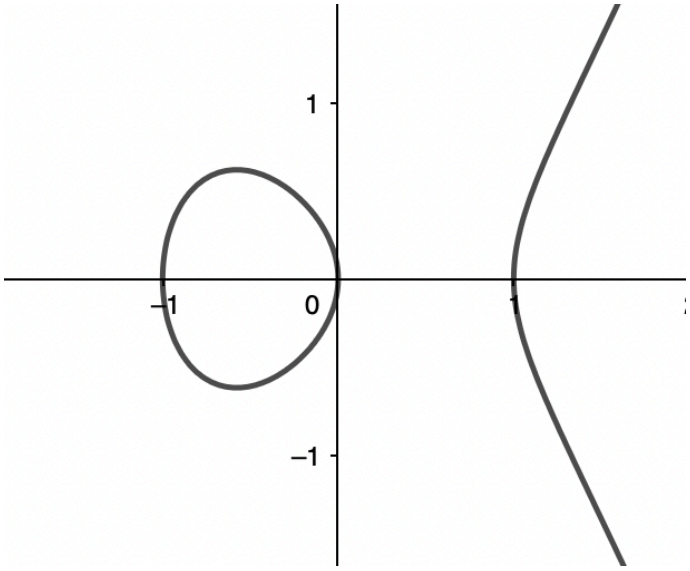
6. Riemann Surfaces I

30 juni

Motivational Question Describe the surface of all complex points on a smooth elliptic curve in $\mathbb{CP}^2$. For instance

$$y^2 = x^3 - x.$$

Is it connected? Is is orientable? How many holes does it have?



Definition. En planär graf är en graf som kan ritas på papper utan att några kanter skär varandra. I en planär graf så kommer kanterna ringa in avgränsade områden, så kallade ytor.

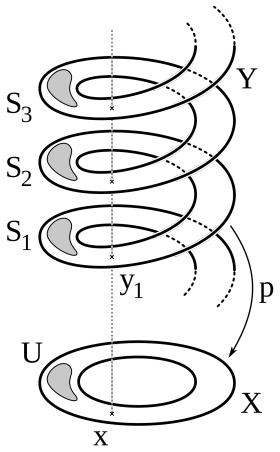
Sats 6.1. **Eulers formel** I en sammanhängande planär graf så är $v-e+f=2$, är v är antalet noder, e är antalet kanter och f är antalet ytor (eller områden). Notera att även den oändliga ytan som omger hela grafen räknas som en yta. En viktig poäng är att de flesta planära grafer kan ritas på flera olika sätt, eftersom en graf inte har något att göra med hur vi ritar den, och vilka ytor vi får varierar. Eulers formel gäller dock för alla möjliga sätt att rita en planär graf så länge inga kanter korsar varandra.

1. Visa att $v-e+f$ inte ändras när man lägger till en kant i en planär graf. Bevisa med hjälp av denna observation, eller på något annat sätt, Eulers formel.

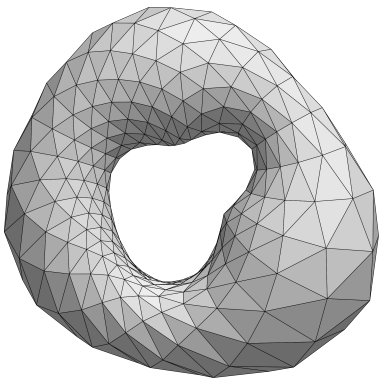
2. Give an example of a graph that is not planar.

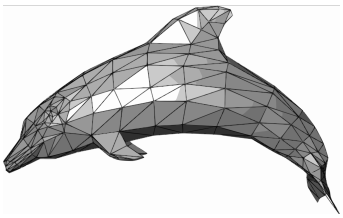
3. Does Euler's formula work, or can you find a similiar formula on

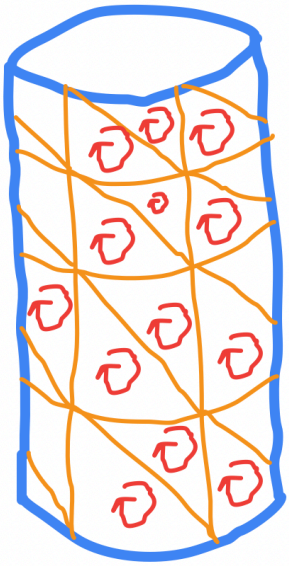
- A sphere
- A Möbius strip
- A klein bottle
- A torus
- A n-torus (torus with n holes)
- (*) The helix as shown to the right



Definition. A surface is called orientable, if it has a triangulation: A decomposition into triangles such that each edge on a triangle is glued to at most one other edge. Each triangle is oriented by choosing a direction around the perimeter of the triangle, associating a direction to each edge of the triangle. If this is done in such a way that, when glued together, neighboring edges are pointing in the opposite direction.







4. Which surfaces from problem 3 are orientable?

5. Is our cubic curve connected?

Definition. On an oriented surface with a sufficiently dense graph, the number $1-(v-e+f)/2$ is called the genus, g.

6. Intuitively, the genus is the number of holes in a surface. Check this, for a torus with n holes.

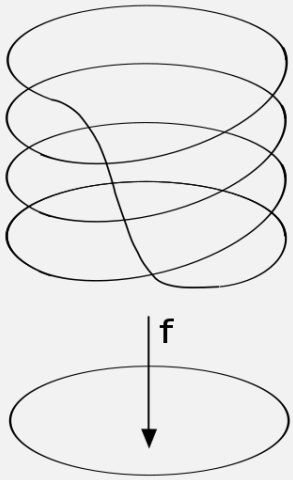
7. What is the genus of our cubic curve?

8. (*) Let Q_1 and Q_2 be two smooth quadric surfaces in $\mathbb{P}^3$. Assume that their intersection Q_1 is a smooth curve. Calculate the genus of this curve.

7. Riemann Surfaces II

2 juli

Definition. A covering map of degree n is a continuous map $f : X \to X$ such that for all points $p \in X$ there are exactly n distinct points $p_1, p_2, \ldots, p_n$ such that $f(p_i) = p$ for $i = 1, 2, \ldots, n$. The following is an example of a covering map of degree 4 from a circle to itself.



1. Which of the following surfaces have a covering map of degree 2? What about degree n?

- A torus
- A Klein bottle
- A Möbius strip
- A sphere
- A torus with 2 holes
- A torus with m holes
- Come up with a surface not on this list

2. How many covering maps can you find of a torus, of degree 2? How many of degree n?

In the last lesson, we discussed that the genus (number of holes) of a smooth cubic curve in the complex projective plane $\mathbb{P}^2$, say $y^2z = x^3 + xz^2$ is 1. The idea was to project the curve onto the $x-$axis to triangulate the cuve, and see that we have Euler characteristic

$$V - E + F = 0.$$

3. Identify a smooth cubic curve with a torus. Describe the covering map of degree n from problm 1, as a map from a cubic curve to itself.

4. How can you find the genus of other curves, say

$$y^2z^3 = x^5 - 3x^3z^2 + xz^4.$$

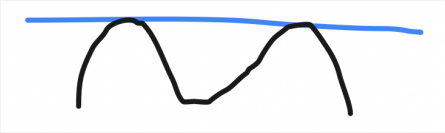
5. What is the genus of a smooth curve of degree d in $\mathbb{P}^2$? How does this result compare to problem 1?

6. What is the genus of a smooth curve of degree d in $\mathbb{P}^2$ with n singular points?

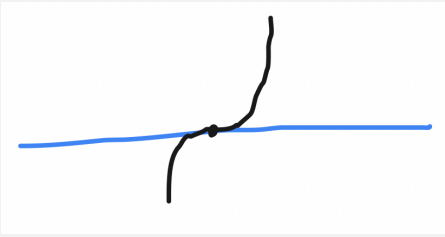
8. Riemann Surfaces III

3 juli

Definition. A bitangent, is a line that is tangent to a curve in multiple points.



Definition. An inflection, is a point where the tangent intersects with multiplicity at least 3.



1. What kind of singular points does these types of tangent lines correspond to in the dual curve? Can a curve have infinitely many inflection points or bitangents?

2. If a smooth curve X has degree d, what is the degree of it's dual? (*) What if X has singular points?

3. (Less important, can be skipped) How can you generalize the genus theorem

$$g = \frac{(d-1)(d-2)}{2}$$

to curves with singular points?

4. How many inflection points does a smooth curve of degree d have?

5. How many points p, such that $p+p+p=0$ does an elliptic curve

$$y^2 = x^3 + ax + b$$

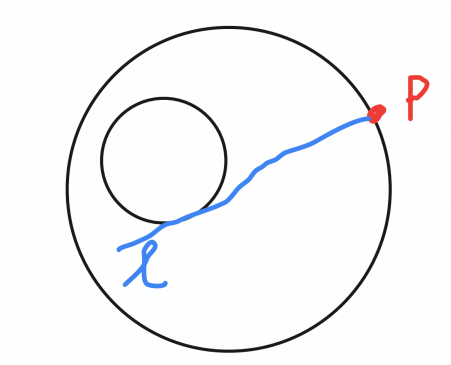
have? Prove that these lie on some lines forming a gitter. (The group operation is defined by defining $p+q$ to be the reflection across the $x-$axis of the 3rd intersection point between the curve and the line through p and q).

6. How many bitangents does a smooth curve of degree d have?

7. Let X be a smooth curve of degree d, and consider a point O on X. Show that O lies on $(d+1)(d-2)$ tangents of X, not counting the tangent at O.

8. Let Ω_1, Ω_2 be circles, conics, or whatever. How can pairs (p, ℓ) where $p \in \Omega_1, p \in \ell$ and ℓ is tangent to Ω_2 - be considered as a cubic curve? Find an explicit equation

on the form $y^2z + x^3 + axz^2 + bz^3$. Can you describe the group operation on the pairs of points and lines?



9. Let T be a cubic curve. Show that there exists, up to isomorphism, 3 curves T' such that there exists an unbranched double-cover $T' \to T$. Prove that T' is a cubic curve. Can you construct it explicitly?

9. Förutspå framtiden (datalab)

2 juli

```
"""
Mattekollo 2022
Datalab i statistik
av Benjamin Verbeek

Denna fil öppnar och läser in data från filen "co2data_2009-2019.txt".
Filen innehåller tre kolumner av data:
    - år (yr) [på decimalform]
    - co2-mängd (co2)
    - månad (month)

Din uppgift är nu att analysera datan.
Uppgifterna står efter dataimporteringen.
Som ledning har du tillgång till en statistikföreläsning från Uppsala
Universitet. Hitta informationen du behöver! Sålla bort det mesta.
Och som vanligt: Fråga om du fastnar!

*** Under vissa uppgiftsnummer står referenser till föreläsnings-pdf:en
och vilken slide som innehåller användbar information. ***

Googla och fråga gärna!
"""

# Importera användbara bibiliotek
import matplotlib.pyplot as plt
import numpy as np

# definiera datafilens namn
filename = "co2data_2009-2019.txt"
# öppna filen
with open(filename, 'r') as f:       # 'r' står för read
    data = f.readlines()             # läs in alla rader (lagras i en lista)
    """
    Programmet går igenom alla rader i listan data, delar på raden, tar
    datan på plats 0, 1, 2 och gör om till float och sedan en numpy-array.
    """
    # konvertera kolumner till numpy arrays
    yr = np.array([float(line.split()[0]) for line in data])    # yr = år
```

```
    co2 = np.array([float(line.split()[1]) for line in data])   # co2 = co2-mängd
    month = np.array([float(line.split()[2]) for line in data]) # month = månad

################################################################################
# Potentiellt användbara funktioner:
#   # np.polyfit(x, y, deg) : anpassar ett polynom av grad deg till x och y.
#                             Returnerar koefficienter i en lista.
#   # np.polyval(coef, x)   : evaluerar ett polynom med koefficienterna coef
#                             vid x. Returnerar en lista med y-värden.
#   # np.linalg.inv(A)      : returnerar inversen av matrisen A.
#   # @ - operatorn         : matrismultiplikation med numpy.
#   Ex: Lös Ax = b  ==>  x = np.linalg.inv(A) @ b

# Plotta gärna dina resultat, t.ex. med hjälp av några av dessa funktioner:
#   # plt.figure(num) : skapar en figure med nummer num
#   # plt.plot(x, y, '-o', label='data')    : plottar x och y med markör
#                                             '-o' och label 'data'
#   # plt.plot(x, y_fit, '-', label='fit')
#   # plt.bar(x, y, label='bar plot')       : plottar x och y som ett
#                                             stapeldiagram
#   # plt.axhline(y=5, color='r', linestyle='--') : plottar en streckad horison-
#                                                   tell linje vid y=5, färg röd
#   # plt.legend()                          : plottar en ruta med namnen på
#                                             plottarna (enligt label)
#   # plt.title('titel')                    : skriver en titel på figuren
#   # plt.savefig('filnamn.png')            : sparar figuren som filnamn.png
#   # plt.show()                            : visar figuren
################################################################################

########## UPPGIFTER ##########
# (a) Ta bort trenden från datan
co2_fit = np.polyfit(yr, co2, 1)

plt.figure()
plt.plot(yr, co2, '-o', label='data')
trend_eval = np.polyval(co2_fit, yr)
plt.plot(yr, trend_eval, '-', label='fit')

co3 = co2 - trend_eval  # removes linear trend

plt.figure()
plt.plot(yr, co3, '-o', label='data')
plt.show()
```

```
# (b) Ta bort säsongsvariationen (gör en rimlig uppskattning av perioden)
#     KOM IHÅG: den resulterande datavektorn bör vara kortare än den
#     ursprungliga. Detta är viktigt om du vill plotta mot år.

co4 = co3[12:] - co3[:-12]
plt.figure()
plt.plot(yr[12:], co4, '-o', label='data')
plt.show()

# Om du har ont om tid (<40 min kvar), hoppa över denna uppgift.
# (c) Autokorrelera datan (korrelera med sig själv, skiftat stegvis)
#     (gamma i föreläsningen) och normalisera gamma för att få fram
#     autokorrelationen rho. Plotta gärna.
# SLIDE 11-12
gammas = np.correlate(co4, co4, mode='full')/len(co4)
gamma_0 = gammas[len(co4)-1]
rho = gammas/gamma_0

N = len(co4)
confidence_limits = [2/np.sqrt(N), -2/np.sqrt(N)]

plt.figure()
plt.bar(range(len(co4)), rho[len(co4)-1:])
plt.axhline(y=confidence_limits[0], color='r', linestyle='--')
plt.axhline(y=confidence_limits[1], color='r', linestyle='--')
plt.show()


"""OBS! Om du har ont om tid: hoppa över (d) och (e) och tjuvkika på den
 färdiga plotten  i fuskmappen."""
# (d) Konstruera rho-matrisen beräkna PACF-koefficienterna phi (se föreläsningen)
# SLIDE 14-15
rho = rho[len(co4)-1:]  # bort med speglingen

# deduce coefficients phi_i of the AR process from the autocorrelations rho_i
rho_matrix = np.array([np.roll(rho, i) for i in range(len(rho))])
rho_matrix = rho_matrix.reshape(len(rho), len(rho))

# upper triangular part is correct now. Copy it to lower part.
rho_matrix = np.triu(rho_matrix)    # first clear lower triangular part.
rho_matrix = rho_matrix + rho_matrix.T - np.diag(np.diag(rho_matrix))

# hitta phi
phi = [(np.linalg.inv(rho_matrix[:i,:i]) @ rho[1:i+1])[i-1] for i in range(1, len(r
```

```
# (e) Plotta phi mot förskjutningen från 0 i ett stapeldiagram, tillsammans
#     med det linjer som markerar det 95%-konfidensintervallet
#     [2/np.sqrt(N), -2/np.sqrt(N)] (N = antalet datapunkter)
#     Identifiera vilka trender som kvarstår i datan: vad påverkar de
#     senare datapunkternas värden?
plt.bar(range(1,len(phi)+1), phi, label=r'$\phi_{ii}$ PACF', width=0.5)
plt.legend()
plt.title(r'PACF')
plt.axhline(y=confidence_limits[0], color='r', linestyle='--')
plt.axhline(y=confidence_limits[1], color='r', linestyle='--')
plt.show()


# (f) Konstruera en modell utifrån de största värdena på phi du hittade
#     med hjälp av psaudoinvers, och plotta den modellen tillsammans med
#     datan.
# SLIDE se exempelfigur på slide 16, modellekvation på SLIDE 17
A = np.array([co4[2:-1], co4[1:-2]]).T

phi_fit = np.linalg.inv(A.T @ A) @ A.T @ co4[3:]

co5 = A @ phi_fit

plt.figure()
plt.plot(yr[3+12:], co5, '-', label='co2 data model prediction')
plt.plot(yr[12:], co4, '-+', label='co2 data', linewidth=0.5)
plt.show()

# (g) Plotta residualerna från datan och modellen.



# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# (*f) Shameless plug: !!! SÖK TILL FYSIK- OCH ASTRONOMILÄGER I HÖST !!! #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

10. Lagrangemekanik I

3 juli

En alternativ metod för att hantera komplicerade problem i fysiken är Lagrangemekanik. Metoden beskrivs av ***Hamiltons princip*** och de resulterande rörelseekvationerna kallas ***Lagranges ekvationer***.

Introduktion av konceptet

Hamiltons princip

Av alla möjliga vägar längs vilket ett dynamiskt system kan röra sig från en punkt till en annan inom ett bestämt tidsintervall (i enlighet med eventuella begränsingar), så är den faktiska vägen den som minimerar[1] *tidsintergralen av differensen mellan kinetisk och potentiell energi.*

1. Föreslå en matematisk beskrivning av Hamiltons princip.

Definition. Vi kallar den kinetiska energin T och potentiella energin U. Vi definierar lagrangianen L som

$$L = T - U$$

2. Givet en partikel med rumskoordinat x, vad beror L typiskt på?

Euler-Lagrange och Lagranges ekvationer

Definition. ***Euler-Lagrangeekvationen*** har lösningar som är stationära punkter för en given verkan.

$$\frac{\partial L}{\partial x_i} - \frac{d}{dt}\frac{\partial L}{\partial \dot{x}_i} = 0, \quad i = 1,2,3 \tag{10.1}$$

Dessa ekvationer utgör ***Lagranges rörelseekvationer.***

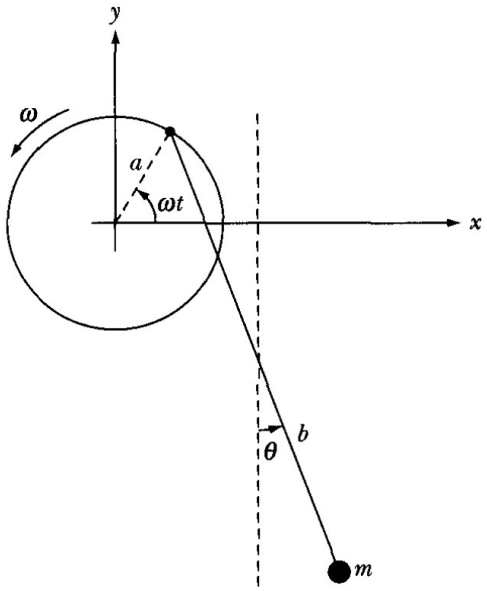
3. Visa att ekvation **10.1** ger samma resultat som newtonsk mekanik för en harmonisk oscillator.

4. Visa nu att även rörelseekvationen för en plan pendel matchar newtonsk fysik. *Betrakta pendelns rörelse i termer av utslagsvinkeln θ, och behandla den som en rektangulär koordinat.*

[1] Egentligen behöver vägarna ej ge ett minimum, utan det kan även röra sig om terasspunkter (och rent formellt även maxima; detta förekommer dock inte i fysiken).

Mer komplexa problem

5. Betrakta nu en simpel pendel av längd b, vars fästpunkt rör sig längs en masslös ram med radie a som roterar med konstant vinkelhastighet ω. Hitta ett uttryck för de kartesiska komponenterna av hastighet och acceleration för massan m. Hitta även vinkelaccelerationen för vinkeln θ (se figur).



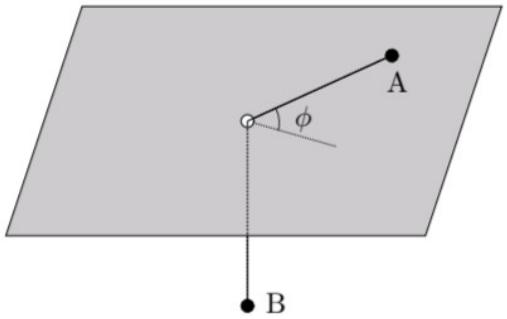
6. En partikel A med massan m_1, rör sig på ett glatt horisontellt plan. Partikeln är fäst i en viktlös otänjbar tråd som löper friktionsfritt genom ett hål O i planet. Tråden uppbär i sin andra ände en partikel B med massan m_2. B rör sig rätlinjigt i lodrät riktning.

Beteckna avståendet OA med x och vinkeln mellan OA och en fix linje i planet med ϕ. Betrakta x och ϕ som (generaliserade) koordinater.

(a) Ställ upp uttrycken för systemets kinetiska och potentiella energi.

(b) Bestäm systemets rörelseekvationer.

(c) För att härleda en i partikeldynamiken känd lag, integrera en av rörelseekvationerna. Vilken lag har du härlett?

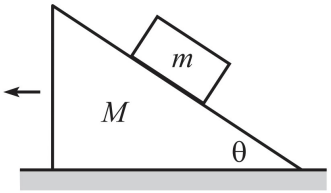


11. Lagrangemekanik II

4 juli

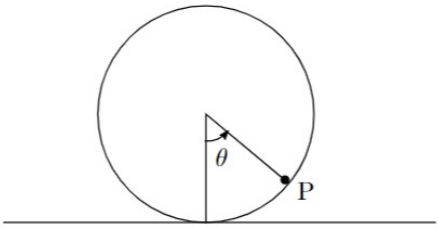
Om du inte redan gjort det, lös gärna problem **6** från gårdagens lektion.

1. En låda med massa m glider friktionsfritt nedför ett plan med vinkel θ. Detta plan har massa M och glider i sin tur friktionsfritt på golvet. Beskriv systemets rörelse.



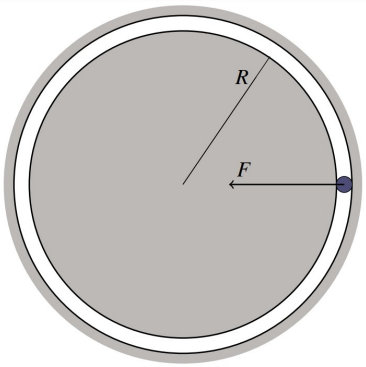
2. Hitta frekvensen för små oscillationer hos en simpel pendel i en tågvagn som har en konstant acceleration a i x-riktningen. Hitta även jämviktsvinkeln θ_e.

3. Ett rakt cirkulärt cylindriskt skal (massa M och radie R) kan rulla utan glidning på ett horisontalplan. Inuti skalet glider en partikel P (massa m) friktionsfritt. Systemet börjar sin rörelse utan begynnelsehastighet med $\theta = \pi/2$. Beräkna cylinderaxelns förflyttning från startläget vid den fortsatta rörelsen som en funktion av θ.



Ledning: Prova att integrera bägge sidor för att få en lösning för x, θ istället för tidsderivator av x, θ.

4. En partikel rör sig i en cirkelbana med radie $r = R$. Bestäm kraften på partikeln från banans väggar genom att lägga till en potential $V(r)$.



TOPOLOGI OCH PARTY-TOPOLOGI

JOHAN BJÖRKLUND

Uppgift 1. *Klassificera "tjocka" bokstäver upp till kontinuerlig omformning. Vad skiljer dem åt?*

Uppgift 2. *Kan du hitta på en sammanhängande "bokstav" som inte kan omformas till någon i vårt vanliga alfabet?*

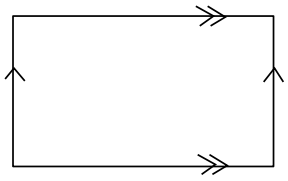
Uppgift 3. *Kan du hitta på oändligt många nya bokstäver så att de parvis är olika?*

Uppgift 4. *Klassificera släta kurvor i planet upp till kontinuerlig omformning (med ett visst antal komponenter, dvs ett gäng cirklar i planet). Upp till 4-5 cirklar kanske?*

Uppgift 5. *Vilka invarianter kan du hitta? (Exempelvis antalet cirklar är en. Omkretsen på den största cirkeln är* inte *en invariant.)*

Uppgift 6. *Om ni kan grafteori: Varje samling av n släta kurvor i planet motsvarar ett (rotat) träd med $n+1$ noder. Vad är kopplingen? Om du inte kan grafteori så spara uppgiften till nästa gång.*

Uppgift 7. *Klassificera släta kurvor upp till kontinuerlig omformning (med ett visst antal komponenter)* **(a)** *på sfären* **(b)** *på en torus. Skillnad mot planet? "Olika" typer av cirklar?*



En modell för en torus är att man tar en rektangel och klistrar ihop sidorna så pilarna matchar. Sedan så klistrar man ihop långsidorna så även de pilarna matchar (se bild). Detta tillåter oss att rita kurvor på en torus lättare.

Uppgift 8. *Fungerar korrespondensen i uppgift 6 även på en sfär eller en torus? Kan du hitta någon korrespondens om det inte fungerar?*

Uppgift 9. *Vad händer om vi vänder på någon pil (t.ex. vi vänder på en av pilarna på kortsidorna). Då kanske vi får en modell för något annat. Ändras vilka kurvor vi kan rita? Samma fråga om vi vänder på både en kortsidepil och en långsidepil. Går det att klistra ihop fysiskt? Försök visualisera vad som händer.*

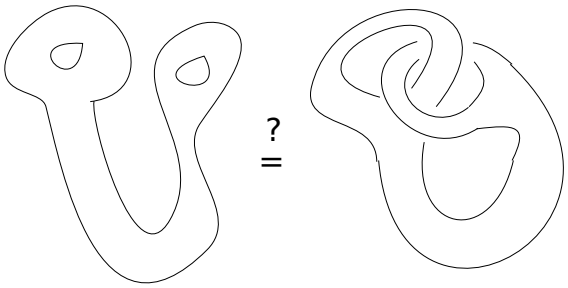
Uppgift 10. *En klassisk uppgift i grafteori är att vi har tre hus och tre resurser (t.ex. vatten, ström,internet) så att vi vill koppla varje hus till varje resurs med en ledning. Ledningar får inte korsa varandra. Detta problem saknar lösning (och visar att det finns grafer som inte går att bädda in i plan). Undersök om problemet går att lösa på någon av våra nya ytor.*

Uppgift 11. *Om vi har fyra hus och fyra resurser, går det att hitta en yta där problemet är lösbart?*

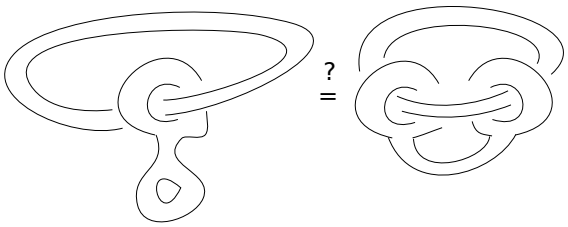
Uppgift 12. *Ett litet flygplan med röd högervinge och grön vänstervinge flyger omkring på en av modellerna i den tidigare uppgiften. Den flyger rakt ut genom en av pilkanterna och tillbaka till mitten. Händer något med vingarna? Kolla för de olika ytorna (torus, en pil vänd, två pilar vända) och de olika kanterna den flyger ut genom (lång/kort).*

PARTY-TOPOLOGI

Uppgift 13. *Handklovs-problemet (om du inte löst det än).*



Uppgift 14. *Visa att ovanstående ytor är lika i meningen att de kan deformeras till varandra i rummet. (De får inte skära sig själva vid kontinuerlig omformning i rummet)*



Uppgift 15. *Visa att ovanstående ytor är lika i meningen att de kan deformeras till varandra i rummet. (De får inte skära sig själva vid kontinuerlig omformning i rummet)*

Uppgift 16. *En morgon så har topologen Tobias tagit på sig sina byxor ut och in. Tobias vill vända tillbaka byxorna så att de blir rättvända, men orkar inte lyfta på fötterna. Kan Tobias göra detta (hans byxor är gjorda av mycket tänjbart gummi eftersom han är topolog)?*

Uppgift 17. *Valentina har med sig en badboll (som också är gjord av topologigummi) till stranden, men upptäcker att den är ut och invänd. Hon kan lätt vända den ut och in genom att använda uppblåsningshålet (dvs en sfär med ett litet hål kan vändas ut och in). Johan har med sig en badring istället, som också råkar vara ut och invänd. Kan han också vända tillbaka sin badring?*

Uppgift 18. *Vi vill hänga upp ett halsband på två spikar så att halsbandet inte trillar ner. Vi vill hänga det på ett sådant sätt att om vi drar ut någon spik så måste det trilla ner, vilken spik vi än väljer.*

(a) *Hur hänger vi halsbandet?*

(b) *Gör samma sak för* 3 *spikar (dvs om vi drar ut en spik så trillar allt)!*

(c) **Gör samma sak för* n *spikar (dvs om vi drar ut en spik så trillar allt)!*

(d) **Gör samma sak för 3 spikar, men så att allt trillar om vi drar ut 2 spikar (vilka vi än väljer).*

(e) **Fundera på någon generalisering!*

(f) **Problemet med ett gäng generaliseringar beskrivs i https://arxiv.org/pdf/1203.3602.pdf för den intresserade.*

Uppgift 19. *Vi har* 4×4 *spikar tillsammans med ett antal ledningar så att varje spik är kopplad till de närliggande spikarna (ej diagonalt). Vissa av ledningarna är trasiga. Du är intresserad av om man kan ta sig från varje spik till en annan spik via icke-trasiga ledningar. Till din hjälp har du en maskin som givet två spikar avslöjar om du kan ta dig mellan dem. Vilket är det minsta antalet mätningar du behöver för att vara säker på att alla spikar är kopplade till varandra?* $2m \times n$ *spikar?* $m \times n$*?*

Harnacks olikhet

En (ickesingulär) kurva C i $\mathbb{C}P^2$ som definieras av ett reellt polynom (som exempelvis $x^2 = yz$) består av dels en del $\mathbb{R}C$ som ligger i det reella projektiva planet $\mathbb{R}P^2 \subset \mathbb{C}P^2$ (där alla koordinater är reella) och dels en komplex del $\mathbb{C}C = C \setminus \mathbb{R}C$.

Uppgift 20. *Om kurvan C är av grad d, vilken genus har den?*

Uppgift 21. *Visa att $\mathbb{C}C$ består av högst 2 delar. Tips: komplexkonjugering.*

Uppgift 22. *Visa att en reell fjärdegradskurva i reella projektiva planet kan bestå av max 4 delar.*

Uppgift 23. *Hur många delar kan $\mathbb{R}C$ högst bestå av om C är av grad d?, Detta kallas Harnacks olikhet.*

Uppgift 24. *Hur kan en reell fjärdegrads kurva i reella projektiva planet se ut?*

Uppgift 25. *Antas alltid max i Harnacks olikhet? Hur konstruerar vi kurvor på ett bra sätt?*

Uppgift 26. * ***Hilberts sextonde problem, löst av Gudkov.*** *Hur kan en reell sjättegrads kurva i reella projektiva planet se ut?*

Uppgift 27. ** ***Olöst senast jag kollade.*** *Hur kan en reell åttonde-grads kurva i reella projektiva planet se ut?*

13. Mattedrabbning blå

5 juli

1. Klara underhåller sig själv på fritiden med att räkna. Hon börjar med ett positivt heltal. Sedan ökar hon talet med dess största äkta delare. Sedan ökar hon det nya talet med dess största äkta delare och så vidare. Visa att Klara efter några sådana operationer får ett tal som är delbart med 3^{2000}.

2. Låt A vara en punkt i planet, och ℓ en linje som ej går genom A. Evan har ingen linjal, utan istället en speciell passare som kan rita en cirkel genom tre distinkta icke-kollinjära punkter. (Cirkelns centrum markeras inte i denna process). Vidare kan Evan markera skärningspunkterna mellan två ritade objekt och markera en godtycklig punkt på ett på ett objekt eller i planet. Kan Evan konstruera punkten där normalen från A till ℓ skär ℓ.

3. På ett plan finns några åttor (som en åtta räknas två cirklar som tangerar varandra i en punkt). Åttorna skär inte varandra på något sätt. Visa att de är ändligt eller uppräkneligt många.

4. I triangeln ABC är vinkeln A lika med 60°. Punkten T ligger inuti triangeln så att $\angle ATB = \angle BTC = \angle CTA = 120^\circ$. M är mittpunkten på sidan BC. Visa att $TA + TB + TC = 2AM$.

5. Tre punkter väljs slumpmässigt på en cirkelperiferi. Beräkna sannolikheten att de tre punkterna bestämmer en spetsvinklig triangel.

6. Med ett *nummer* menar vi en sifferssträng med sex siffror från 000000 till 999999 (totalt finns det en miljon nummer). Ett *turnummer* är ett nummer där summan av de första tre siffrorna är lika med de sista tre. Hur många turnummer finns det?

IV Programmering - GPT1

1. Programmering med legorobotar, del 1

28 juli 2022

Idag ska vi gå igenom vad en robot är, hur programmeringsmiljön ser ut och skriva våra första program. Vi ska gå igenom hur man kan komma åt sensorerna och motorerna och skickar över sina program. Vi kollar även på hur man får roboten att låta, och skriver ut saker på robotens skärm.

1.1 Uppgift: Åka i en fyrkant

Programmera roboten att åka i en kvadrat, fast med en loop. Här ska ni loopa 4 iterationer. Vad är fördelarna med att använda en loop istället för att skriva ut alla looparna?

Vad som kan varieras

Storleken på fyrkanten. Antal varv man ska åka runt i fyrkanten. Enkelt att ändra bara en siffra istället för att lägga till 8 rader kod för varje varv.

Var används detta på riktigt?

Programmerare är lata. Genom att använda loopar slipper man mycket arbete. Det är bättre att programmet automatiskt upprepar saker istället för att man ska skriva många rader kod.

1.2 Uppgift: Lagerrobot

1.2.1 Beskrivning

Placera ut lite läskburkar som roboten ska ta tag i och flytta. Hämta en burk och lämna den bakom linjen till exempel. Här måste ni använda mediummotorblocket för att styra klon.

1.2.2 Vad som kan varieras

Man kan testa att ändra hastighet som klon stängs eller öppnas. Ska man använda tid eller rotationer eller grader här? Förklara varför!

1.2.3 Bakgrund

En robot måste manipulera verkligheten för att få kallas robot, annars är det bara en maskin (ett annat krav är att roboten ska kunna reagera på sin verklighet). Nu får du lära dig att programmera roboten till att utföra sysslor istället för att bara åka omkring. Det kan t.ex. vara att hämta läsk från kylen när man sitter framför TVn.

Lagerrobotar runt om i världen arbetar såhär. De plockar upp ett objekt och transporterar det från punkt A till punkt B. Det kommer att revolutionera trans-

portindustrin. En taxibil behöver ingen förare om den klarar av att köra mellan olika adresser på egen hand.

1.3 Inbrottslarm

1.3.1 Beskrivning

Programmera ett inbrottslarm som väntar på att något ska komma framför US (**u**ltra**s**onic, ultraljud) sensorn och sedan tjuter!

1.3.2 Vad som kan varieras

Låt roboten aktivera klon samtidigt som den låter. Så kan den slå till inbrottstjuven! Testa olika tröskelvärden på avståndssensorn.

Se hur den kan övervaka en dörr och reagera när någon försöker ta sig in. Försök att smita förbi larmet utan att bli upptäckt. Testa att sätt flera robotar på olika höjd (en på golvet, en på en stol, en på ett bord och en på en stol på ett bord) och ta sig förbi. Sätt en på en stol på ett bord och försök köra virtuell limbo: Försök ta er under den osynliga ultraljud-strålen utan att utlösa larmet! Vem klarar det?

1.3.3 Bakgrund

Inbrottslarm! Behöver jag säga något mer?

1.4 Uppgift: Åk nära hinder

1.4.1 Beskrivning

Ställ upp ett hinder. T.ex. en kartong. Roboten ska starta bakom en starten startlinje riktad mot hindret. Du ska programmera roboten att komma så nära hindret som möjligt utan att välta det med hjälp av en ultraljudssensor. Roboten ska stanna när den har något framför sig. Använd dig av vänta-på-ultraljud-blocket.

1.4.2 Vad som kan varieras

Avståndet innan roboten stannar. Experimentera gärna på hur långt som sensorn klarar som mest respektive minst. Notera att sensorn har ett minimumavstånd som den inte kan se kortare än för inget ljud ekar tillbaka till receptorn. Varje sensor är unik, men runt 5cm. Maxavståndet är 255cm, och allting längre än så kommer säga 255cm. Små objekt på långt avstånd är svåra att upptäcka med ultraljud. Testa hur små objekt som kan detekteras och på vilka avstånd!

1.4.3 Bakgrund

Ni har byggt en robot som reagerar på ultraljud. En människa kan inte höra ultraljud, men en fladdermus kan. En fladdermus navigerar med hjälp av ultraljud precis på samma sätt som er robot. Den skickar ut en högfrekvent signal och mäter tiden det tar för ekot att komma tillbaka, med lite matte kan fladdermusen då räkna ut hur långt det är till kvistar och grenar i skogen.

1.5 Uppgift: Patrullrobot

Programmera en robot att åka fram och tillbaka mellan två svarta streck på marken för evigt.

Vad som kan varieras

Istället för att använda streck kan man ha en ultraljudssensor som känner av väggar. Roboten kan alltså åka mellan de två väggarna i korridoren utanför klassrummet och patrullera.

Var används detta på riktigt?

Om en robot ska vakta ett industriområde (eller en dörr) behöver den åka fram och tillbaka mellan två markeringar och ha koll på vad som händer. Robotar gör ofta väldigt enformigt arbete som människor inte vill göra.

1.6 Uppgift: Styra roboten

Ge roboten möjlighet att bli styrd med hjälp av dess knappar på EV3-klossen. Trycker man på pil uppåt ska roboten åka ca 10cm framåt. Trycker man på bil bakåt ska den åka bakåt. Pil höger roterar den 90 grader höger och pil vänster blir 90 grader åt vänster.

Vad som kan varieras

Om man trycker på mittenknappen ska klon öppnas om den är stängs, och stängas om den är öppen.

Var används detta på riktigt?

Ibland klarar inte en robot av att ta rätt beslut. Roboten kanske hamnar i en situation som den aldrig tidigare varit i och vet inte vad den ska göra. Då är det bra om en människa kan gå in och styra roboten rätt. Det kan också vara bra om roboten gör något farligt och man måste manuellt flytta tillbaka roboten, för ofta är de för tunga för att flyttas av muskelkraft.

1.7 Uppgift: Städrobot

Hålla sig inom ett område och städa rent det. Varje gång roboten ser en linje ska den backa, svänga och köra framåt igen. Puttar ut alla läskburkar från området tillslut.

Vad som kan varieras

Storleken på området. Man skulle också kunna lägga till något så att programmet avslutas när roboten städat ett tag (begränsar antal loop-iterationer).

Var används detta på riktigt?

Kan användas för att städa"ett bord (puttar ned allt på golvet). Ungefär som en städrobot eller dammsugarrobot. Gräsklipparrobotar brukar ha en linje/slinga runt hela trädgården som roboten ska hålla sig innanför.

1.8 Uppgift: Dammsugarrobot

Undvika att köra in i något. Samma som uppgift 1.7 fast man väntar på att roboten ska känna av något med ultraljudssensorn istället.

Vad som kan varieras

Området som ska köras runt. Kan vara bara på golvet och undvika väggar och kartonger och stolar. Dock har ultraljudssensorn svårt att känna igen smala bords- och stolsben.

Testa att svänga ett slumpmässigt antal grader vid varje sväng.

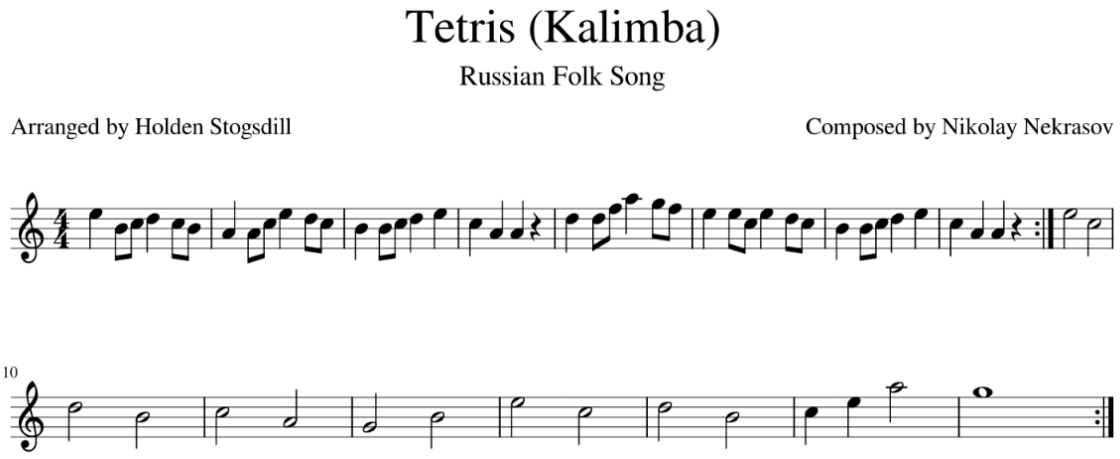
Var används detta på riktigt?

Detta är på samma sätt som robotdammsugare fungerar. En robotdammsugare krockar i saker och märker det och backar och svänger. Er robot är till och med bättre eftersom den inte behöver slå i någonting, den reagerar innan den krockar i nått!

Vanliga dammsugarrobotar är optimerade för att så snabbt som möjligt täcka av ett rum. Men eftersom roboten inte på förhand vet hur rummet ser ut används ofta slumpen och statistiska metoder för att öka chanserna att snabbt städa hela rummet.

1.9 Uppgift: Tetris

Om du kan läsa noter kan du låta roboten spela upp den ryska folkvisan som används i spelet tetris.



Vad som kan varieras

Testa att spela in en annan låt.

2. Programmering med legorobotar, del 2

29 juni 2022

Idag kommer vi fördjupa oss i programmeringsstrukturer, till exempel loopar, trådar och if-satser. Vi kommer även att skriva egna funktioner med in-argument.

2.1 Uppgift: Linjeföljare

Följa en linje. Bryt ned enkla mänskliga instruktioner följ en linjetill en sekvens av instruktioner som en dator förstår. Det är inte det enklaste att förstå hur roboten ska följa en linje.

Vad som kan varieras

Hur linjen ska dras. Testa att göra olika skarpa linjer eller korsningar och se vad som händer.

Var används detta på riktigt?

Linjeföljande robotar finns överallt! Det finns tävlingar med dem och de används i industrin. Många robotar guidas av kablar. Vår robotgräsklippare hittar tillbaka till laddstationen med hjälp av en guidekabel som den följer på samma sätt som ni precis gjort. En förarlös bil ser linjerna på vägen och följer dem.

2.2 Uppgift: Följa linje till burk

Beskrivning

Följ linjen tills roboten hittar en burk. Fånga den och sen fortsätta följa linjen.

Vad som kan varieras

Istället för att fortsätta följa linjen med burken kan man fånga den vrida roboten 90 grader och köra av linjen och släppa burken, för att sen backa tillbaka till roboten känner av linjen och vrida upp sig och fortsätta följa linjen!

Man kan ställa dit ett hinder längsmed linjen som roboten ska detektera och köra runt, och sedan fortsätta på andra sidan.

Man kan också låta linjen upphöra på ett ställe och fortsätta längre fram. Då ska roboten upptäcka att linjen tog slut och sedan fortsätta följa den längre fram. Detta är en svår uppgift att lösa med endast en ljussensor, men det går!

Var används detta på riktigt?

Robotar på ett pappersbruk förflyttar stora pappersrullar genom att följa linjer på golvet.

Det är ofta det kommer fram en människa framför transportroboten och då måste roboten såklart kunna stanna! Annars dör människan. Ännu bättre vore förstås om roboten kunde köra runt människan.

2.3 Uppgift: Sluta följa linjen när den kommer till vit markering

Sluta följa linjen när den kommer till en vit markering.

Vad som kan varieras

Man kan med fördel placera en burk i slutet av linjen som roboten ska gripa tag i och förflytta!

Var används detta på riktigt?

Robotar använder sig ofta av något som kalla beacon, vilket betyder markörer på svenska. Dessa använder roboten för att veta var den befinner sig. De kan bestå av t.ex. färgmarkeringar eller saker som sänder ut strålning som roboten reagerar på.

2.4 Uppgift: Egen funktion för att köra rakt fram

Skapa en egen funktion som tar in cm som roboten ska åka istället för varv eller grader på hjulen. Använd variabler för hjulens storlek.

Vad som kan varieras

Bygg vidare på funktionen för att köra rakt och lägg in ett gyro så att den alltid kör rakt oavsett om det är ojämnt underlag.

Var används detta på riktigt?

Bra med funktioner som man kan återanvända fler gånger. Dessutom bra att kunna ange i enheter som är enklare att relatera till. Skapar mindre fel då.

2.5 Uppgift: Egen funktion för att svänga

Skapa ytterligare en funktion som tar in antal grader roboten ska svänga. Testa att gör den **utan** gyro-sensorn, utan med matematik som räknar ut hur långt motorerna ska svänga för att uppnå en viss vinkel.

Vad som kan varieras

Bygg vidare på funktionen ovanför och ange en svängradie på svängen, istället för att alltid svänga kring robotens centrum. Då kan man ange hur stor cirkel robotens centrum ska förflytta sig längsmed.

2.6 Uppgift: Övergångsställe

Räkna linjerna i en variabel, och stanna efter att roboten passerat 4 linjer. Skulle man vilja ändra nått ljusvärde behöver man bara göra det på ett enda ställe.

Vad som kan varieras

Kör fram till en vägg och räkna antalet linjer som roboten har passerat fram tills dess. Skriv ut antal passerade linjer på skärmen.

Var används detta på riktigt?

Linjerna på rad kan ses som ett övergångsställe som roboten måste upptäcka.

Variabler är bra för att kunna komma ihåg saker som har skett tidigare.

2.7 Uppgift: Håll avståndet!

Försök att hålla ett visst avstånd till ett föremål framför den, t.ex. din hand eller en kartongskiva. Roboten ska backa om kartongen kommer för nära, och köra framåt om kartongen flyttas bort från roboten.

Vad som kan varieras

Olika avstånd. På hur långt avstånd kan den maximalt följa med? Vad händer om du rör föremålet för snabbt? Vad händer om den tappar bort föremålet? Lägg till någon form av sökbeteende? Kan ni sätta flera robotar i rad som följer varandra?

Går att skapa en P-regulator, vilket betyder att den åker fortare om kartongen är långt ifrån gränsen, och långsammare om den är nära avståndsgränsen.

Var används detta på riktigt?

Volvo utvecklar just nu fordonståg där lastbilar kör utan förare. Det finns en förare i första lastbilen som styr, sedan lägger sig flera andra på ett fast avstånd till bilen framför. Detta för att det blir billigare att slippa betala lön till förarna i lastbilarna längre bak, dessutom spar de bränsle eftersom de kan ligga väldigt tätt och få bort luftmotstånd.

2.8 Uppgift: Gyro-regulator

Gyroregulator, får roboten att åka rakt med hjälp av data från gyro-sensorn. Det här är vad som brukar kallas P-regulator.

Vad som kan varieras

Försök få roboten att klara så stora störningar som möjligt. Lägg ut papper framför roboten som gör att hjulen spinner/slirar. Eller lägg ut andra små störningsmoment, kanske tändstickor om vi har några. Knuffa på roboten.

Lägg till ett ljud som säger Äj"eller "uschöm nån puttar på roboten för mycket.

Utveckla P-regulatorn mer och lägg till en integrerande och deriverande del så att det blir en PID-regulator.

Var används detta på riktigt?

Att köra rakt trots störningar är viktigt. Världen är sällan helt ideal och ska man göra en robust robot måste den använda sig av sensorer för att ta sig fram. Om man bygger en skogsrobot som ska ta sig fram över stockar och stenar används ofta gyrosensorer.

2.9 Uppgift: Grafritare

Låt roboten mäta avståndet framför roboten eller vinkeln på roboten och rita ut sensorvärdet som en graf på skärmen. Du kan både låta roboten rotera själv, eller lyfta upp roboten och rotera med handen.

Vad som kan varieras

Istället för att mäta avstånd kan man plotta något annat. T.ex. färgen på underlaget så får du en "karta"över vilka nyanser golvet har längsmed en sträcka. Förslagsvis låter man roboten köra framåt istället för att rotera på stället.

Var används detta på riktigt?

Ofta så får man en bättre förståelse av sensordata i en graf istället för att bara läsa av en siffra på en skärm. Det är ofta viktigt att se hur något varierar över tid och i en graf kan du se historisk data enkelt.

2.10 Uppgift: Kartering

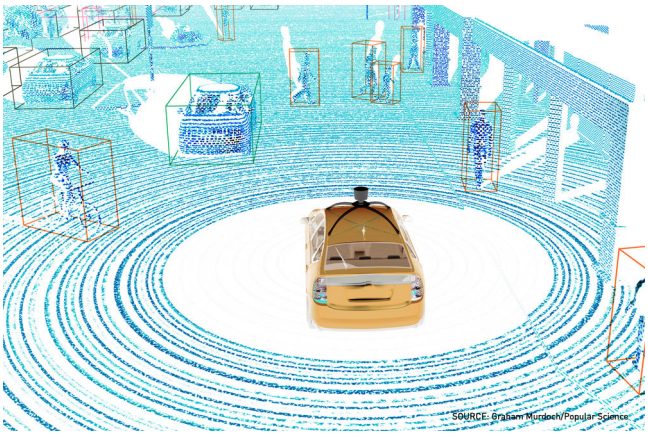
Låt roboten rotera ett varv och mät avståndet fram till närmsta objekt vid varje grad. Rita ut en graf i fönstret på roboten där X koordinaten representerar vinkel och Y-koordinaten motsvarar dess avstånd. Då får man en form av karta över omgivningen.

Vad som kan varieras

Istället för att bara rotera ett varv kan man låta roboten köra framåt eller bakåt och sen rotera igen för att bygga en större karta. Detta kallas för SLAM.

Var används detta på riktigt?

Räddningsrobotar skickar man in i okända områden och då måste de skapa sig en karta när de åker runt. De använder ultraljud och lasersensorer (LIDAR) för att mappa upp området som de är aktiva i.



Figur 2.1: En förarlös bil bygger upp en karta av sin omgivning.

3. Programmering III: Python-introduktion

30 juni

3.1 Datatyper

Hur man deklarerar variabler med olika datatyper:

```
x = 7                                       # Integer
y = 3.7                                     # Float
z = 'Text here'                             # String
bl = True                                   # Bool
lst = [53, False, ['Hugo', 4, 'orange']]    # List
dct = {'key1':'value1', 'key2':'value2'}    # Dictionary
st = {'banana', 13, 'apple'}                # Set
tpl = (5, 7, 9)                             # Tuple
```

3.2 Interaktivt läge

I interaktivt läge kan man skriva kod rad för rad som sedan tolkas av datorn och exekveras (körs) direkt.

3.2.1 Miniräknare

Starta Python i interaktivt läge och testa att skriva in olika numeriska uttryck för att använda terminalen som en miniräknare.

3.2.2 Symbolhanterande

Testa att lagra värden i olika variabler och utför beräkningar. Använd även olika datatyper. Vad blir resultatet av att genomföra följande operationer:

(a) Ett tal plus ett tal?

(b) En sträng plus en sträng?

(c) En sträng plus ett tal?

(d) En sträng multiplicerat med ett heltal?

(e) Ett heltal plus eller gånger en bool?

(f) En lista gånger ett heltal?

(g) En lista plus en lista?

3.2.3 Indexering av strängar

Vad händer när man skriver följande? Fundera först på vad ni tror, innan ni testar. Reflektera och fråga om nått är oklart!

```
>>> x = "Hello world"
>>> x[7]
>>> x[1:3]
>>> x[-2]
>>> x[3:]
>>> x[:] # När vill man använda detta?
>>> x[:1]
>>> x[:-3]
>>> x[-5:]
```

3.2.4 Indexering av listor

Vad händer när man skriver följande? Hur ser a ut efter varje rad? Fundera innan ni skriver in det i terminalen. Ger någon rad error?

```
>>> a = [2,3]
>>> a[1] = 5
>>> a[1] = [3,4,5]
>>> a[2] = [7]
>>> a[2:] = [7]
>>> a[2:] = [7,8,9,10]
>>> a[2:] = [3,4]
>>> a[:-3] = [0]
```

3.3 Skript

Oftast så skriver vi inte kod interaktivt utan sparar den i en textfil. Detta gör att vi enkelt kan återanvända stora mängder kod, till exempel om man vill utföra liknande beräkningar många gånger med olika parametrar, eller vill testa att något fungerar utan att behöva skriva om alla test mellan varje ändring.

3.3.1 Skapa ett skript

Skriv ett program som utför ett par beräkningar och skriver ut svaret.

Extra: Be användaren om in-data till beräkningarna. **Extra:** Loopa beräkningen till dess att användaren vill avsluta programmet.

3.4 Funktioner

Ofta nöjer vi oss inte med de inbyggda funktionerna utan vi vill kunna definiera egna funktioner:

```
def add(a, b)
    return a + b
```

Implementera någon funktion som tar ett eller flera argument och returnerar ett lämpligt svar. Ett exempel är att konvertera längder i fot och tum till cm. 10’5” betyder 10 fot och 5 tum. En fot är 12 tum och en tum är 2,54 cm. Anropet length("10’5”") ska alltså returnera 317,5. Glöm inte att testa olika specialfall som enbart tum och enbart fot! Om du blir klar snabbt kan du lägga till att användaren via en parameter får bestämma om konverteringen ska ske från tum och fot till cm eller tvärtom.

3.4.1 Rekursion

Rekursion innebär att en funktion kan kalla på sig själv.

```
def rec(count)
    if count > 0:
        print('There are ' + count + 'function calls left!')
        return rec(count - 1)
```

Skapa en egen rekursiv funktion. Exempel:

1. n! (n-fakultet)
2. fibonacci-talen
3. Ackermann-funktionen (Googla!)

Extra: Lägg till en variabel som räknar hur många rekursiva anrop som görs till funktionen. Finns det smartare sätt att implementera din funktion som inte kräver lika många anrop?

3.5 Indexering

Python är ett nollindexerat språk. Det innebär att man börjar räkna på 0. Till exempel name[0] innebär den första bokstaven i strängen name, name[1] är den andra bokstaven osv. Negativa index räknar bakifrån, name[-1] är det sista tecknet i strängen. Detta skrivsätt fungerar även på till exempel listor.

3.5.1 Extra: Slicing

Python har väldigt smidig syntax för att extrahera intervall. I exemplet ovan så betyder name[2:5] strängen med bokstäverna som har index 2, 3 och 4 (men inte 5). Om man skriver name[:-2] får man alla tecken utom de 2 sista.

3.6 Extra: Comprehension

Python har flera olika sätt att med kort och koncis kod skapa de objekt man vill. Till exempel finns list comprehension, set comprehension och dictionary comprehension. Kolla upp vad dessa innebär. Ser du varför den här notationen är användbar?

3.6.1 Övningar listor

Använd list comprehension (listbyggare) för att lösa följande uppgifter:

1. Skapa en lista med alla kvadrattal mellan 1 och 100.
2. Skapa en lista med alla tal mellan 1 och 100 som innehåller siffran 6.
3. Utgå från listan ['Björn', 'Hund', 'Katt', 'Häst']. Lägg till ett utropstecken till alla strängar och filtrera bort strängar som inte innehåller bokstaven 'n'.

3.6.2 Övningar ordböcker

Använd dicct comprehension (ordboksbyggare) för att lösa följande uppgifter:

1. Skapa en dictionary vars nycklar är talen från 1 till 15 (båda inkluderas) och värdena är kvadraten av nycklarna. Testa först med en for-loop och sen med en ordboksbyggare.
2. Skapa en dictionary med alla orden i denna delfråga där orden är nyckeln och ordens längd är dess värden. Använd ordboksbyggare.
3. Använd nästlade list / dictionary comprehensions för att hitta största ensiffriga delaren för varje tal från 1 till 1000. Talen är nyckel och största ensiffriga delaren är värdet.

3.7 Extra: Klasser

Python används ofta för att skriva objektorienterad kod. Detta innebär att man kan skriva egna klasser, med egna medlemsfunktioner. Implementera en egen klass! Ett exempel finns nedanför:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def celebrate_birthday():
        self.age += 1

    def present_themself():
        print('Jag heter ' + self.name + ' och jag är ' + self.age + ' år!')

p1 = Person('Hugo', 22)
p1.present_themself()
p1.celebrate_birthday()
p1.present_themself()
```

3.8 Fler instruktioner och exempel

`https://docs.python.org/3/tutorial/introduction.html`

4. Tävlingsprogrammering

2 juli

Idag ska vi gå igenom hur tävlingar fungerar i programmering och öva på ett par problem. Vi kommer även gå igenom hur man använder kattis.

Problem

1. Skriv ett program som avgör hur många gånger talet x finns med i en lista. Ditt program bör läsa in x med hjälp av input() funktionen och skriva ut svaret med hjälp av print(). Listan kan för tillfället kodas direkt i programmet men bör vara lätt att ändra på.

   Kontrollera erat program med följande exempel på in- och utdata.

   - Lista = $[1,2,3,4]$, x = 2 -> svar = 1
   - Lista = $[1,1,2,2]$, x = 1 -> svar = 2
   - Lista = $[0,1,2,3]$, x = 5 -> svar = 0
   - Lista = $[0,1,0,1]$, x = 1 -> svar = 2

2. Uppdatera nu programmet så att ni läser in listan med hjälp av input(). Ett vanligt sätt som man gör detta på i tävlingar är att första läsa in hur lång listan är och sen läsa in innehållet. Som exempel kan man då dela upp listan $[1,2,3,4]$ på fem rader. Första raden blir då en 4a följt av talen 1, 2, 3 och 4 på en rad var.

3. Skriv ett program som avgör ifall det finns två tal i en lista som summerar till x. Programmet ska först läsa in talet x och sedan läsa in listan på samma sätt som i förra uppgiften. Ifall det finna två tal i listan som summerar till x, ska programmet skriva ut ”Ja”. Annars ska programmet skriva ut ”Nej”.

   Testa programmet med ett par exempel som du hittar på själv.

4. Skriv ett program som avgör ifall det finns tre tal i en lista som summerar till x. Läs in input på samma sätt som tidigare. Ifall det finns tre tal som summerar till x, skriv ut ”Ja”, annars ”Nej”.

   Testa programmet med ett par exempel som du hittar på själv.

5. Hur många högertrianglar finns det vars hypotenusas längd är ett heltal mindre än 100? Skriv ett program som testar alla möjliga kombinationer på triangellängder. För varje kombination, testa ifall talen följer pythagoras sats. Skriv ut antalet kombinationer med hjälp av print().

Kattis

6. Vi ska nu övergå till att göra några problem på kattis. Öppna upp `open.kattis.com` i din browser. Tryck på log in högst upp till höger. Välj sedan ett av alternativen (vi föreslår log in with email). Ifall ni har ett konto sedan tidigare, logga in på

det kontot. Annars måste ni skapa ett. Tryck på sign up for a Kattis account och fyll i all information.

7. Börja med problemet Hello world: `https://open.kattis.com/problems/hello`. Skriv först programmet i en fil på eran dator. När ni har skrivit programmet kan ni skicka in det till kattis genom att först trycka på upload files... och sedan submit.

8. Lös https://open.kattis.com/problems/planetaris

9. Lös https://open.kattis.com/problems/harshadnumbers

10. Lös https://open.kattis.com/problems/ostgotska

Lite svårare problem

11. Finns det något tresiffrigt tal som är 20 gånger större än sin egen siffersumma? Skriv ett program som beräknar svaret. Ifall det finns sådana tal, skriv ut alla. Ifall det inte finns några, skriv ut ”finns inga”.

12. Skriv ett program som avgör ifall det finns någon delmängd av en lista som summerar till x. Läs in x samt en lista med input() och skriv ut ”Ja” ifall det finns en delmängd vars summa blir x. Skriv annars ut ”Nej”.

    Tips: Börja först med en lista av längd 2. Öka den sedan med 1 i taget upp till 10 så att du gradvis får ett svårare problem.

13. Givet en lista och ett heltal x, vad är den delmängdsumma som är närmast x utav alla möjliga kombinationer. Precis som i förra uppgiften kan det vara smart att först lösa problemet för väldigt små listor och sen jobba sig uppåt.

5. Pygame och Objekt

5 Juli

5.1 Objekt och klasser

1. Skapa en klass `Rabbit` med parametrarna `firstname`, `lastname` och `age`.
2. Skapa en metod till personklassen för att hälsa ger en text där haren introducerar sig.
3. Lägg till metoden `birthday` som ökar en harens ålder med 1.
4. Lägg till funktionen som gör att `str()` fungerar på objekt av klassen `Person`.
5. Lägg till metoden `__add__()` till klassen som beskriver vad som händer när två kaniner blir tre. (Den ska skapa en ny kanin med ett slumpmässigt namn från en lista och åldern 0)

5.2 Kom igång med PyGame!

Följande kod kan användas för att starta pygame och öppna ett vitt fönster:

```
import pygame

WINDOW_HEIGHT = 800
WINDOW_WIDTH = 800

pygame.init()

screen = pygame.display.set_mode([WINDOW_WIDTH, WINDOW_HEIGHT])

while True:
    screen.fill((255,255,255))

    pygame.display.flip()
```

Du kan antingen utgå ifrån den tomma koden ovan och hitta på något eget. Exempelvis så skulle du kunna skapa en animation av olika former som studsar mot skärmens väggar.

Du kan också bygga vidare på mitt exempel från genomgången. Du kan ladda ned det från denna länken: `https://drive.google.com/drive/folders/1m_O-Eook_4Vi5tFidQjMs-D_4FEsGMXN?usp=sharing`

Förslag på vad som går att lägga till på exemplet:

1. Bättre ”Game Over” skärm
2. Bakgrund

3. Fler spikar för ökad svårighetsgrad
4. Ökande hastighet på spikarna
5. Poängsystem
6. Objektiv, exempelvis något som spelaren ska plocka upp för poäng
7. Möjlighet att starta om spelet
8. Ljudeffekter
9. Nya fiender
10. Möjlighet att hoppa
11. Något helt eget!

V

Programmering - GPT2

1. Inbyggda Python-funktioner

28 juni 2022

Lambda

1. Skriv en lambdafunktion som tar in x som parameter och returnerar x+2, tilldela funktionen till variabeln L:

```
i=6
L=
print(L(i))
```

2. Skapa en lambda-funktion som du skickar in till key-parametern i sort() funktionen för att sortera listan i fallande ordning. Normalt sett sorterar sort() i stigande ordning. Det finns en reverse-parameter som man normalt sett använder för att byta ordning, men den får du inte använda i denna uppgift.

```
numbers=[100, 10, 10000, 1, 9, 999, 99]
```

3. Sortera en lista med postkod-ortnamn tupler efter ortens namn:

```
orter = [(18452, 'Österskär'), (58334, 'Linköping'), (12152, 'Johanneshov'),
    (75642, 'Uppsala')]
```

Zip och map

4. Zippa ihop listorna med postorter och postnummer så de hamnar i en lista av tupler.

```
numbers=[18452, 58334, 12152, 75642]
cities=["Österskär", "Linköping", "Johanneshov", "Uppsala"]
```

5. Använd zip, list och sort för att skapa en dictionary där strängarna och siffrorna matchar upp efter hur de sorterats.

```
strings=["Fredrik", "Daniel", "Elias", "Hugo"]
values=[4, 12, 7, 19]
res == {"Fredrik": 12, "Daniel": 4, "Elias": 7, "Hugo": 19}
```

6. Använd map för att returnera alla kvadrater av talen i listan:

```
numbers = [4,8,2,9,1]
```

7. Använd lambda och map för att addera ihop elementen i första och andra listan. Listorna är alltid lika långa. Tips: Lambda kommer ha två argument.

```
first = [4,8,2,9,1]
second = [3,6,7,0,1]
```

8. Använd map(), lambda och count() för att skapa en ny lista som innehåller antalet a:n (både stora och små) i varje delstat:

```
states=["Alaska", "Alabama", "Arizona", "Arkansas", "Colorado", "Montana",
    "Nevada"]
res == [3, 4, 2, 3, 1, 2, 2]
```

9. Skriv ett program som räknar ut summan av absolutbeloppen i listan. Använd sum och map.

```
lst=[99.3890,-3.5, 5, -0.7123, -9, -0.003]
```

List- och dict-comprehensions

10. Skapa en lista med hjälp av list-comprehensions med 6 adderat till varje elem:

```
lst=[44,54,64,74,104]
```

11. Använd list comprehensions för att bygga en lista med kvadraterna av varje tal vars kvadrat är större än 50.

```
lst=[2, 4, 6, 8, 10, 12, 14]
```

12. Skapa en lista (med list comprehension) med alla fordonsnamn utifrån nedanstående dictionary. Ta bara med fordonen som väger mindre än 5000kg, och passa även på att konvertera namnen till VERSALER i samma list comprehension.

```
vehicles={"Sedan": 1500, "SUV": 2000, "Pickup": 2500, "Minivan": 1600, "Van":
    2400, "Semi": 13600, "Bicycle": 7, "Motorcycle": 110}
```

13. Använd dictionary comprehension för att skapa en ny dictionary där endast key-value paren med börsvärde över 2000 finns med.

```
stocks={"NFLX":4950,"TREX":2400,"FIZZ":1800, "XPO":1700}
```

Filter

14. Använd filter för att skapa en lista med bara de jämna talen:

```
lista = [4,7,9,1]
```

15. Använt filter() och list() för att filtrera ut en lista med vokalerna från:

```
str1="Mattekollo 2022 arrangeras på Valla folkhögskola."
```

16. Använd map och filter för att lägga till 2000 till de tal som är under 8000.

```
lst1=[1000, 500, 600, 700, 5000, 90000, 17500]
```

2. Spelprogramering 2022

29 juni

Idag kommer vi att gå igenom python-biblioteken PyGame som man kan använda för att skapa spel eller rita annan grafik eller simuleringar på datorn.

Spelprogrammering - PyGame

1. Öppna ett fönster

```
import pygame # Importera PyGame biblioteket

pygame.init() # Starta allt internt i PyGame

# Välj storlek på fönstret med bredd (800) och höjd (600).
screen = pygame.display.set_mode([800, 600])

while True:
    # Render
    screen.fill((255, 255, 255)) # Fyll skärmen med vit

    # Uppdatera det som visas. Alla tidigare kommandon ritar till
    pygame.display.flip()

# Rensa allt som skapats vid init()
pygame.quit()
```

2. Fönstret går i nuläget bara att stänga genom att skriva *CTRL+C* i terminalen. För att programmet ska stoppa när man trycker på stäng av knappen måste det kodas. Allt användaren gör, tex trycker på tangenter eller stäng-krysset läggs till som ett event som koden kan reagera på.

```
...
running = True
while running:
    # Events
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
```

3. Rita former

```
    ...
    # Render
    screen.fill((255, 255, 255))

    pygame.draw.rect(screen, (255, 0, 255), (100, 200, 500, 300))
    # Rita en lila rektangel, men övre hörnet i punkten (100, 200) (100 från
```

```
        vänster, 200 uppifrån) som är 500x300 pixlar stor

    pygame.display.flip()
    ...
```

Alla olika draw funktioner står här: `https://www.pygame.org/docs/ref/draw.html`. Det finns bland annat för cirklar och linjer.

4. Interaktivitet

(a) Testa om en tangent blivit nedtryckt:

```
    ...
    # Event
    for event in pygame.event.get():
        ...
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_a:
                print("A was pressed this frame")
        if event.type == pygame.KEYUP:
            if event.key == pygame.K_a:
                print("A was released this frame")
    ...
```

Alla olika K_ värden finns en bit ner på denna sida: `https://www.pygame.org/docs/ref/key.html`.

(b) Gör så att kvadraten flyttas när man trycker på tangentbordet, förslagsvis WASD eller piltangenterna.

5. Gör nu så att kvadraten fortsätter flytta på sig när man kan hålla in tangenterna, så att man inte måste klicka flera gånger för att flytta den långt.

Detta kan till exempel göras med `pygame.key.get_pressed()`. Den returnerar en sekvens av bools, en för varje tangent som kan indexeras med K_ konstanten.

```
pygame.key.get_pressed()[pygame.K_a]
```

6. Begränsa till ett lägre framerate och introducera delta tid.

`clock.tick(FPS)` returnerar hur många millisekunder som har passerat sedan förra uppdateringen, det delat på 1000 ger då hur många sekunder. Om tiden som har gått är mer än $\frac{1}{\text{FPS}}$ kommer `clock.tick(FPS)` att vänta så att spelet inte går snabbare än FPS ggr per sekund.

```
...
clock = pygame.time.Clock()
...
dt = 0
while running:
    ...
    square_x += 10 * dt # Ändra variablen square_x med 10/sek
```

```
    ...
    dt = clock.tick(60) / 1000.0
```

7. Ladda in och rita bilder

```
# Ladda in bilden, om den har en transparent bakgrund använd .convert_alpha()
name_pic = pygame.image.load("name.png").convert()
# Skala om bilden till rätt storlek.
name_pic = pygame.transform.scale(name_pic, (400, 400))
...
    # Render
    ...
    # Placera bilden på skärmen, med övre vänstra hörnet i punkten (10, 10).
    screen.blit(name_pic, (10, 10))
```

Byt ut spelaren (tidigare boxen) till en kanin (eller annan valfri bild).

8. För att integrera musen kan följande användas:

```
        ...
        if event.type == pygame.MOUSEBUTTONDOWN: # eller MOUSEBUTTONUP
            if event.button == 1: # 1 = vänsterklick, 3 = högerklick
                print(f"Left click was pressed this frame, at {event.pos}")
        ...

    # Få mus-positionen (0, 0) är övre vänstra hörnet.
    pos = pygame.mouse.get_pos()
    print(f"Musen är på x: {pos[0]}, y: {pos[1]}.")

    # Få musens delta-rörelse sedan förra uppdatering
    rel = pygame.mouse.get_rel()
    print(f"Musen flyttade sig {rel[0]} pixlar i x och {rel[1]} pixlar i y.")
```

Gör så att man kan placera en morot genom att klicka i spelet. Gärna så att mitten av moroten placeras där man klickar.

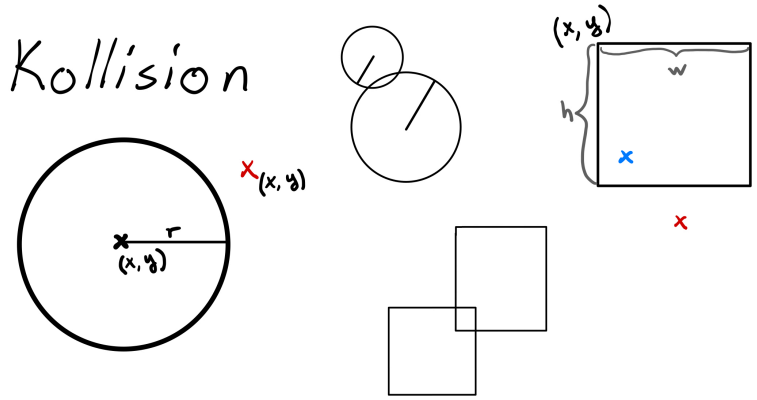
9. Kollision

Skriv funktioner som kan testa kollision i dessa fyra basfal.

- Kollision mellan cirkel och (punkt | cirkel)
- Kollision mellan rektangel och (punkt | rektangel)

Implementera sedan att kaninen kan äta morötterna genom att gå in i dem.

Man kan också testa dessa kollisioner med PyGames Rect. Det är ett smidigt sätt att ha en rektangel som man kan flytta runt och testa kollisioner med: `https://www.pygame.org/docs/ref/rect.html`.



10. Score och text

Lägg till en score som ökar för varje morot som äts. Visa den på skärmen genom att rita text.

```
pygame.font.init()
my_font = pygame.font.SysFont('helvetica', 20)
...
    # Render
    ...
    # Skapa en bild med texten 'Poäng: 10', med anti-aliasing, som är svart
    text = my_font.render("Poäng: 10", True, (0, 0, 0))

    # Rita texten i punkten (10, 100)
    screen.blit(text, (10, 100))
```

11. Uppdatera morötterna så att det går att placera ut flera stycken samtidigt.

12. Skapa en funktion som lägger till en morot på en slumpad position på skärmen. Skapa nya morötter då och då automatiskt.

13. Skapa en rival som också försöker äta morötterna. Välj en bild, få den att gå mot morötterna och äta dem. Välj om den tex ska gå mot den som är närmast eller räkna ut om spelaren kommer hinna före till en specifik morot och välj dess morots-mål strategiskt.

Det här är grunderna för att göra spel. Du kan antingen jobba vidare på detta. Lägg till hinder som kaninen och rivalen måste undvika. Lägg till olika powerups och gör så att rivalen också kan använda dem. Eller något helt annat!

14. Kodfilen är nog redan nu väldigt lång, och det blir snabbt många variabler att hålla koll på. Att använda klasser och sprites underlättar i stora projekt. Mer info finns bra beskrivet bland annat här: `https://realpython.com/pygame-a-primer`.

3. Pyplot, Numpy och ekvationslösning

30 Juni

Tilläggsfiler finns att hämta här: `http://shorturl.at/eszDZ`
Stjärnmärkta ($\star$) uppgifter kräver lite mer matematik och är tänkta att vara frivilliga.

3.1 Introduktion till Pyplot

1. Plotta funktionen $f(x) = -x^2 + 3x + 1$ för $x \in [-3, 5]$.
2. Plotta funktionerna $f(x) = sin(x)$ och $g(x) = x - \frac{x^3}{6}$ i olika färger för $x \in [-2, 2]$.
Sidonot: Visst ser graferna lika ut för små x?
3. Rita de två föregående problemen som två subplotter ovanpå varandra.

3.1.1 Andra typer av grafer

4. Ett kast med en fusktärning kan i python simuleras med koden nedan. Använd denna kod för att simulera 10000 kast med tärningen och redovisa sedan antalet gånger tärningen landar på respektive sida i ett histogram.

```
import random
import matplotlib.pyplot as plt


def throwDice(): # Kast med fusktärning via invers transform sampling
    # Generera slumptal:
    x = random.random()
    # Sannolikhet för varje sida (sannolikhetsfunktion):
    sannFunk = [0.15, 0.15, 0.15, 0.15, 0.15, 0.25]
    # Kumulativ sannolikhetsfunktion för att hitta träffad sida:
    csannFunk = [sum(sannFunk[0:i+1]) for i in range(0,len(sannFunk))]
    # Hitta vilken sida tärningen landar på
    for i,P in enumerate(csannFunk, start=1):
        if P > x:
            return i

print(throwDice())
```

Tips: Det finns en histogram funktion i matplotlib (googla!) som kan rita det åt dig, men för att göra uppgift 6 lättare rekommenderar jag att använda *ax.bar()* istället.

5. $\star$ Illustrera additionen av vektorerna $\vec{v} = (1,2)$ och $\vec{u} = (2,3)$ med hjälp av graftypen quiver.
Ledtråd: Ni behöver dessa parametrar för quiver, annars kommer pilarna skalas om:

```
ax.quiver(x,y,u,v, angles='xy', scale_units='xy', scale=1)
# x, y är listor med startpunkterna för varje vektor
# u, v är listor med vektorerna
# angles='xy' gör att pilar pekar från (x,y) till (x+u,y+v)
# scale_units='xy' gör att pilarnas längd följer axlarnas skala
# scale = 1 sätter förhållandet till 1:1 mellan vektorer och axlar
```

3.1.2 Animation

6. Animera stapeldiagrammet med tärningskast du skapade tidigare genom att rita in ett par kast åt gången, anpassa tiden per bild efter smak.

7. En massa som hänger i fjäder dras ifrån sitt jämviktsläge och släpps så att massan oscillerar vertikalt. Massans rörelse kan i detta fall beskrivas av $y = -\cos t$, där t är tiden. Representera massan som en stor cirkel och animera dess rörelse. Välj lämplig uppspelningshastighet.

3.2 Numpy

8. Skapa en vektor med tal från 10 till 49. *Ledtråd:* `np.arange`

9. Skapa en 3x3x3 array med slumptal. *Ledtråd:* `np.random.Generator.integers`

10. Skapa en vektor med 10 slumptal och sortera den. *Ledtråd:* `sort`

11. Skapa en 3x3 matris med talen från 0 till 8. *Ledtråd:* `reshape`

12. Hur kan man lägga till en ram (fylld med nollor) runt en existerande array? *Ledtråd:* `np.pad()`

13. Skapa en array med nollor på formen (20,20) med ettor på yttersta kolumnen och raden. *Ledtråd:* `np.zeros()`

14. Hur kan du få tag i (x, y, z) indexet för det hundrade elementet i en array som har formen (6,7,8)? *Ledtråd:* `np.unravel_index()`

15. Vilka av följande operationer går att göra med en numpy array av godtycklig form? Vad gör de? Fundera innan du kör dem.

```
Z**Z
2 << Z >> 2
Z <- Z
1j*Z
Z/1/1
Z<Z>Z
```

16. Skapa en 16x16 array och skapa sedan en ny array utifrån de fyra block summorn (varje block är 4x4). *Ledtråd:* `np.add.reduceat`

17. Skriv om funktionen som räknar fram ett kast med en fusktärning så att den använder numpys funktioner istället, inga loopar ska behövas längre. Använd funktionen `np.random.Generator.choice`, googla hur du använder den!

3.2.1 Inläsning av data

Ladda ner linReg.csv filen med ett antal approximativa punkter för två funktioner $f(x)$ och $g(x)$.

18. Rita upp punkterna från båda funktionerna som prickar.

19. Anpassa en rät linje (1:a grads polynom) till punkterna för funktionen $f(x)$. Rita den anpassade linjen tillsammans med punkterna.

20. $\star$ Vad för typ av funktion passar till punkterna för funktion $g(x)$? Gör en kurvanpassning för bestämma funktionen, du kan behöva bearbeta datan innan. Rita den anpassade funktionen tillsammans med punkterna.

3.2.2 $\star$Linjär algebra

21. $\star$ Beräkna matris-vektor multiplikationen nedan:

$$\begin{pmatrix} 2 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 0 \end{pmatrix}$$

Ledtråd: `@/np.matmul()` (Googla)

22. $\star$ Låt $\vec{v} = (1,\ 3,\ 2)$ och $\vec{u} = (0,\ -2,\ 1)$, beräkna $\vec{v} \times \vec{u}$. (Googla)

23. $\star$ Beräkna determinanten av en slumpmässigt genererad 5x5 matris. (Googla)

24. $\star$ Slumpa fram 3x3 matriser med heltal mellan 0 och 9 tills du får en inverterbar matris (determinanten är nollskild). Beräkna då inversen. (Googla)

25. $\star$ Skapa en klocka med en visare som vrids med ett steg per sekund och gör ett varv på 60 sekunder. Använd plot eller quiver för att skapa visaren och förslagsvis en rotationsmatris för att vrida visaren.

3.3 Ekvationslösning

Definition. *Bisektionsmetoden* är en metod för att hitta det tal x som uppfyller $f(x) = 0$ för en given kontinuerlig funktion f.

1. Betrakta ett kort intervall $[a, b]$ där f byter tecken.
2. Välj en ny punkt i mitten av intervallet $c = \frac{a+b}{2}$.
3. Om c inte är ett nollställe så har $f(c)$ samma tecken som antingen $f(a)$ eller $f(b)$. Välj ett nytt intervall $[a, c]$ eller $[c, b]$ på vilket f byter tecken.
4. Upprepa punkt 1-3 tills nollstället är inom ett tillräckligt litet intervall.

Definition. *Sekantmetoden* är en numerisk metod för att hitta det tal x som uppfyller $f(x) = 0$. Givet två tidigare approximationer, x_n och x_{n-1} så kan en förbättrad approximation x_{n+1} av x^* fås:

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-2})}$$

26. Hitta en rot till funktionen med två korrekta decimaler: $f(x) = 4x^3 + 4x^2 + 6$ med hjälp av bisektionsmetoden.

27. Hitta en av lösningarna till ekvationen: $x^7 + 9x^3 + 9x - 6 = 5x + 5$ med 5 korrekta decimaler.

28. Hitta de två lösningarna som är närmast $x = 0$ till ekvationen $\arctan x = 2\sin(\pi x + 2)$ med 5 korrekta decimaler.

29. Hitta en lösning till ekvationen $-3\ln(\sin 2x) - 2 = 0$, $x \in [-1, 1]$ med 8 korrekta decimaler. Lös först med bisektionsmetoden och sedan med sekantmetoden, jämför antalet iterationer som behöver göras!

30. Hitta lösningen som ligger i närheten av $x = 2$ med 8 korrekta decimaler till $x^4 - 5.39x^3 + 7.2207x^2 + 3.9471x - 9.6636$. Lös först med bisektionsmetoden och sedan med sekantmetoden, jämför antalet iterationer som behöver göras!

31. Ställ upp en ekvation som låter dig approximera $\sqrt{2}$. Beräkna sedan värdet med 12 korrekta decimaler.

4. Tävlingsprogrammering

2 juli

Idag ska vi gå igenom hur tävlingar fungerar i programmering och öva på ett par problem. Vi kommer även gå igenom hur man använder kattis och lära oss om ett par vanliga metoder som är användbara inom tävlingar men även programmering i allmänhet.

Kattis introduktion

1. Till att börja med måste vi skapa ett konto på kattis, då det är där vi kommer att öva på en del uppgifter. Senare under kollot kommer vi även att använda kattis för att hålla en tävling. Öppna upp `open.kattis.com` i din browser. Tryck på log in högst upp till höger. Välj sedan ett av alternativen (vi föreslår log in with email). Ifall ni har ett konto sedan tidigare, logga in på det kontot. Annars måste ni skapa ett. Tryck på sign up for a Kattis account och fyll i all information.

2. Börja med problemet Hello world: `https://open.kattis.com/problems/hello`. Skriv först programmet i en fil på eran dator. När ni har skrivit programmet kan ni skicka in det till kattis genom att först trycka på upload files... och sedan submit.

3. Lös https://open.kattis.com/problems/planetaris

4. Lös https://open.kattis.com/problems/harshadnumbers

5. Lös https://open.kattis.com/problems/ostgotska

Blandade problem

6. Lös: https://open.kattis.com/problems/closestsums

7. Lös: https://open.kattis.com/problems/guess

8. I detta problem kommer vi att läsa in all indata via standard input och skriva ut svaret med print() precis som vi gjorde på kattis.

   - Input: på första raden står ett heltal, $n <= 10^5$. På de följande n raderna står en sträng var som definierar listan s. Sedan kommer en rad med ett heltal, $q <= 1000$, som get antalet frågor. På de kommande q raderna kommer ett ord var, $q[i]$.
   - Output: för varje ord $q[i]$, skriv ut den alfabetiskt minsta strängen från listan s som är större än eller lika med $q[i]$ (dvs hitta det alfabetiskt minsta x från s sådant att $x >= q[i]$).

9. Clara har ett staket som hon vill måla. Staketet består utav n stycken plankor i led och varje planka får endast målas i en färg. Clara har k burkar med olika färger att välja på. Hon får återanvända samma färg så många gånger hon vill

men det får ändast finns ett ställe där två närliggande plankor får samma färg. På hur många sätt kan Clara måla sitt staket?

10. Det finns ett kvadratiskt rutnät av storlekt $n \times n$. Pingvinen Clara står högst upp till vänster i rutnätet och vill ta sig till det undre högra hörnet. För att göra det får Clara gå antingen åt höger eller neråt. Dock får hon inte gå till vänster eller uppåt.

I varje ruta finns det en viss mängd fisk som Clara kommer att äta upp ifall hon passerar rutan. Mängden fisk i varje ruta ges av ett heltal (det kan finnas flera av samma tal i rutnätet). Clara vill veta vad den smartaste vägen till hörnet är, dvs den väg där hon får äta mest fisk. Hur många fiskar får hon äta längs med den smartaste vägen?

11. Pingvinen Clara blev väldigt nöjd efter att hon fått äta massor med fisk. Tyvärr åt hon upp allting för snabbt och är hungrig igen. Av någon anledning befinner hon sig på en gata med n stycken hus i rad. Clara har frågat varje husägare hur mycket fisk som de har i sina kylskåp och har planerat att stjäla så mycket hon kan. Clara kommer att bryta sig in i ett hus i taget och äta upp all fisk hon hittar. Tyvärr kommer ägaren att säga till båda sina grannar så Clara kommer inte lyckas att råna husen precis till vänster eller höger om ett hus hon redan har rånat. Hur mycket fisk kan Clara stjäla ifall hon väljer hus på ett så smart sätt som möjligt?

Skriv en funktion som tar in en lista med heltal. Talet på index i i listan representerar hur mycket fisk som finns i hus i. Funktionen bör returnera svaret till ovanstående problem.

12. Givet två strängar, a och b, hur lång är den längsta strängen som finns som substring i båda strängarna? Skriv en funktion som löser detta problem med hjälp av dynamisk programmering. Erat program bör fungera snabbt, även på strängar som har längd 1000. Testa erat program genom att skapa strängar av längd 1000. Testa dels med strängar som är slumpade, men även strängar som innehåller till exempel bara bokstaven A.

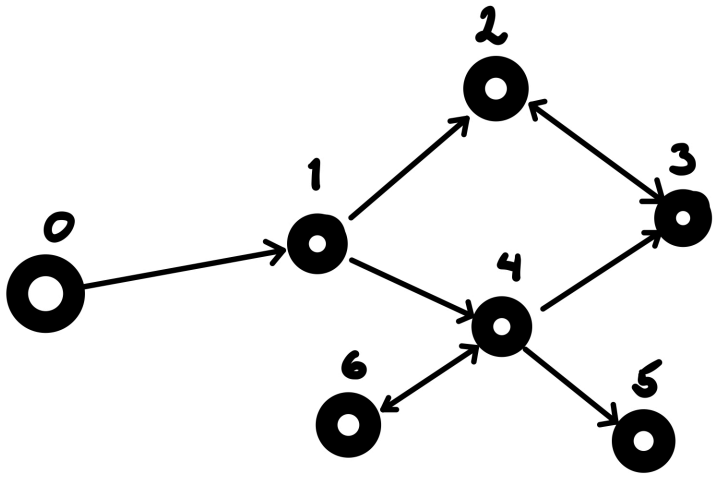
5. Grafteori 2022

5 juli

Idag kommer vi att gå igenom grafteori.

Grafteori 2022

1. Spara grafer: (Föreläsning)



Grannlista (för grafen ovan)

n	grannar
0	1
1	1,4
2	3
3	2
4	3,5,6
5	
6	4

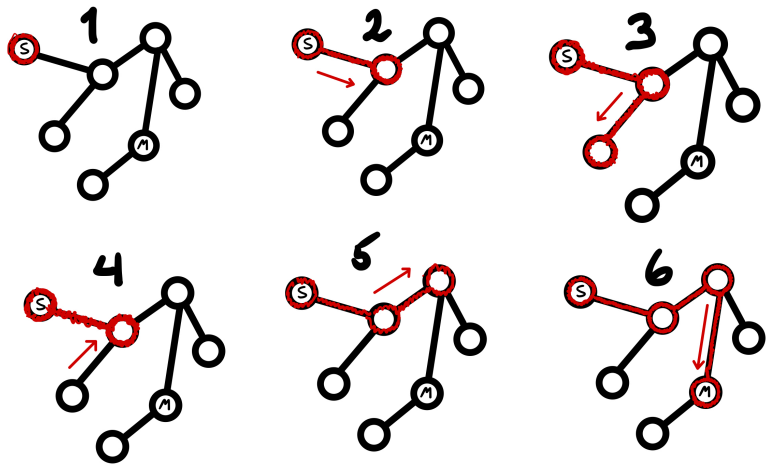
Matris (för grafen ovan om alla vikter är 1)

0	1	0	0	0	0	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	1	1
0	0	0	0	1	0	0
0	0	0	0	0	0	0
0	0	0	0	1	0	0

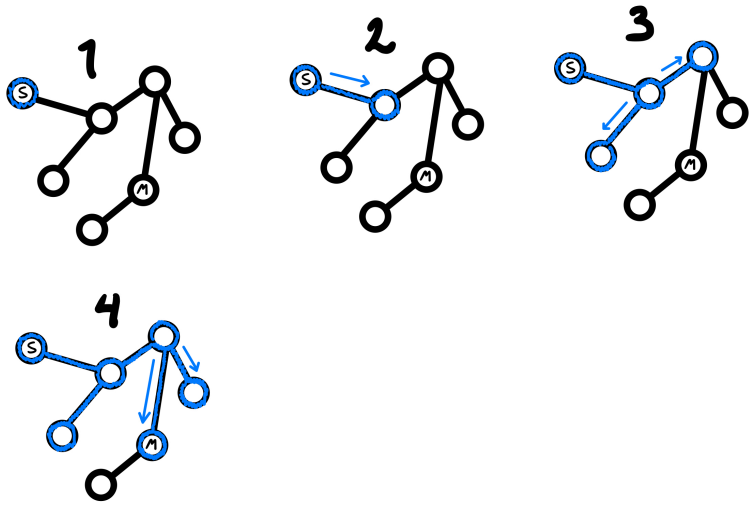
2. DFS och BFS (föreläsning)

DFS och BFS (Djupet- och Bredden först sökning) är två algoritmer för att söka igenom grafer. I en DFS går datorn igenom grafen och fortsätter tills den kommer till en återvändsgränd då den går tillbaka till den senaste korsningen med outforskade vägar. I en BFS går datorn igenom alla alternativ i varje korsning. Se Figur 5.1 (röd) och Figur 5.2 (blå) för exempel på några steg i respektive algoritm i en liten graf.

DFS och BFS har olika styrkor och svagheter, ett tydligt exempel är om man vill hitta den kortaste vägen mellan (S)tart och (M)ål. När BFS har hittat en väg till målet är det garanterat den kortaste och man kan avbryta utan att titta igenom



Figur 5.1: Exempel på DFS

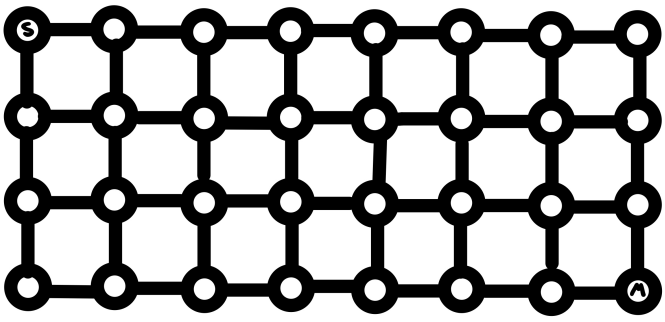


Figur 5.2: Exempel på BFS

resten av grafen. Detta gäller inte för DFS då man måste kolla alla alternativ innan man är säker. Hitta på grafer där detta blir tydligt om det känns oklart.

Implementera en DFS- och en BFS-funktion som söker igenom en graf sparad som en grannlista.

3. Tänk er följande graf:



En DFS skulle vara väldigt långsam då antalet sätt att ta sig från S till M är enormt! (Observera att det ej finns begränsningar mot att röra sig upp eller till vänster). Men med hjälp av dynamisk programmering kan vi göra även en DFS snabb i detta fall. Om vi på varje nod sparar vilken som är den kortaste vägen dit vi har hittat hittills, behöver programmet inte utforska omvägar om det redan har hittat en kortare.

Modifiera er DFS lösning så att den sparar kortaste vägen till alla noder. Dvs ändra på listan med besökta noder från en boolsk array till en array med heltal. Uppdatera överallt där du använder dig av besokt variabeln.

4. Skriv en BFS som tar sig från ordet `gud` till ordet `fan` och som bara får ändra på en bokstav ett steg i taget.

5. Då grafer kan representeras på många olika sätt förväntas man kunna läsa in dem på många olika sätt. Lös dessa kattisproblem för att både öva på grafalgoritmer och att ta in data:

- `https://open.kattis.com/problems/birthday`
- `https://open.kattis.com/problems/flyingsafely`
- `https://open.kattis.com/problems/weakvertices`
- `https://open.kattis.com/problems/grid`
- `https://open.kattis.com/problems/walkforest`

6. Dijkstra algoritmen kan användas om man snabbt vill hitta den garanterat kortaste vägen i en graf med viktade kanter (dvs olika kanter är olika långa). Den går ut på att ha flera aktiva noder som rör sig utåt likt en BFS, men i Dijkstra fortsätter vi på den nod som har kortast väg hittills. Om noden sedan når målet fortsätter algoritmen tills alla andra aktiva noder har färdats längre än den till målet. Det är då garanterat att det är den kortaste vägen.

Hint: För att effektivt hitta nästa nod med minst väg kan `PriorityQueue` användas.

```
from queue import PriorityQueue
pg = PriorityQueue()
pg.put(3) # Lägger till ett element i kön.
pg.put(5)
pg.put(2)
 # Returnerar det minsta elementet som finns i kön och tar bort det.
x = pg.get() # x = 2
y = pg.get() # x = 3
z = pg.get() # x = 5
```

VI Programmering - GPT3

1. Datastrukturer

28 juni

Uppvärmning

1. Skriv en funktion find(lis, x) som returnerar första indexet där x förekommer i listan *lis*. Ifall x inte förekommer i listan, returnera None.

2. Det enda funktionsanroppet du får göra för att ta bort element från en lista är pop(), en gång. Dvs du måste ta bort sista elementet ur en lista exakt en gång, men du får annars inte ta bort ett element mitt i listan. Skriv en funktion remove(lis, i) som tar bort elementet på index *i* ur listan *lis* med ovanstående krav.

Länkad lista

En länkad lista är en lista som består utav enskilda noder, där varje nod innehåller ett värde. För varje nod förrutom den sista i listan, så finns det en länk till nästa nod. Den sista noden länkar istället till None. För varje nod förrutom den första i listan, så finns det en länk till den föregående noden. Den första noden länkar istället till None, då det inte finns nån föregående nod.

3. Implementera en klass, Node, som fungerar som en länkad lista. Klassen bör stödja minst följande funktioner:

- addBack(x)
- addFront(x)
- popBack(x)
- popFront(x)

Ledtråd: varje objekt bör spara två variabler, next och previous, som håller ordningen på listan.

4. Implementera en funktion som tar bort elementet från index i ur listan. Ifall indexet inte är giltigt, gör ingenting (kan förekomma ifall *i* är större än längden på listan).

5. Implementera en funktion som vänder på listan. Undvik helst att skapa nya objekt.

Union Find

Nu ska vi implementera en klass, Union Find. Datastrukturen ska hålla koll på vilka index som tillhör samma grupp och ska gå snabbare än att linjärt söka igenom alla noder vid varje anrop. Minimumkrav på vilka funktioner som bör finnas är:

- size(x)

- find(x)
- sameSet(a, b)
- join(a, b)

6. __init__ funktionen bör ta in en parameter som anger hur många element som finns totalt. I __init__ funktionen ska alla index hamna i egna grupper, och alla storlekar ska vara 1.

7. Implementera funktionen size(x). Den ska returnera hur många element som tillhör samma grupp.

8. Implementera funktionen find(x). Ifall x är ledaren av gruppen, returna x. Sök annars upp ledaren och returnera den.

9. Implementera funktionen sameSet(a, b). Returnera True ifall a och b har samma ledare. Returnera annars False.

10. Implementera funktionen join(a, b). Ifall a och b redan har samma ledare, gör ingenting. Koppla annars ihop grupperna så att den mindre gruppen får den andre som ledare.

11. Uppdatera find(x) funktionen så att den förkortar vägen till ledaren vid varje anrop.

12. Skriv ett program som kan hålla koll på vilka mattekollodeltagare som är med i samma grupp. Principen är densamma som för Union Find, men istället för att två index anges, så kommer deltagarenas namn att anges.

Rödsvart träd

Börja med att implementera ett binärt sökträd. Se till att all funktionalitet fungerar innan du börjar med rödsvarta delen. Nedan följer förslag på en implementation för ett binärt träd:

13. Ha en klass, Tree, som håller koll på lite information så som antalet noder i trädet, en referens till roten och potentiellt mer information om ni vill.

14. Ha även en klass, Node, som håller koll på information för varje nod i trädet. Den bör ha en referens till dens vänstra och högra barn, samt ett värde som noden skall hålla i. Det kan även underlätta att ha en referens till föräldern, då detta gör det lättare att stega sig igenom trädet.

15. Det bör finnas en funtion, insert(value), i Tree klassen som lägger till en nod med motsvarande värde i trädet. Stega er igenom trädet tills ni når botten av trädet och stoppa in en ny nod där.

16. Det bör även finns en funktion, find(value), som letar upp noden som innehåller det givna värdet. Ifall värdet inte finns i trädet så ska None returneras. Annars kan ni antingen returnera True eller nod-objektet.

17. En remove funktion, remove(value), är även bra att ha. Den bör leta upp noden som innehåller ett visst värde, ta bort den, och uppdatera trädet så att det är korrekt.

Bonus till binära trädet

Det finns även vissa andra funktioner som kan vara kul att ha med. I C++ finns det till exempel en funktion som heter lower_bound(x). Denna funktion är lik find, men letar istället upp den nod med lägst värde av dem som är större än eller lika med x.

Man kan också skriva en funktion, getIndex(i), som returnerar noden som skulle ha legat på index i i en sorterad lista.

2. Pussellösare

29 juni

Under dagens lektion ska vi konstruera en pussellösare till ett lämpligt logikpussel. Ni vår välja vilket pussel ni vill lösa, men lättast är förmodligen klassiska pussel så som sudoku. Till att börja med kan det även underlätta att begränsa pusslets storlek (t.ex. sudoku på ett 4×4 rutnät).

Representation av pusslet

1. För att kunna lösa ett pussel krävs först att vi kan representera pusslet i datorn. Välj ett spel och leta upp ett exempel av erat pussel online. Skapa en textfil som innehåller en textversion av pusslets innehåll (tänk på att all information om pusslet måste bevaras i textfilen ni skapar. Dvs ni måste kunna lösa pusslet enbart utifrån den information ni sparar i textfilen).

2. Skriv kod för att läsa in textfilen ni precis skapade. Se sedan till att eran kod lagrar information på ett lämpligt sätt. T.ex. kan ni spara en sudoku som en matris av siffror. Ifall ni har ett mer komplicerat pussel kan ni exempelvis spara mer information med hjälp av en klass.

   Då erat program kanske blir ganska långt under denna lektion kan det vara smart att dela upp programmet i funktioner. Ni måste inte, men det är smart att ha en funktion som sköter inläsningen.

3. Skriv en funktion som skriver ut pusslet på ett fint sätt. Tänk på att det ska vara lätt att läsa, så att ni snabbt kan kontrollera ifall det ser rätt eller fel ut. Ni kommer senare att använda den här funktionen för att skriva ut era lösningar.

4. Testa att erat program hittills fungerar. Kan ni läsa in en textfil med ett pussel och sedan skriva ut samma sak?

Lösaren

Nu är det dags att skriva kod för att faktiskt lösa pusslet. Vi kommer göra detta genom en rekursiv funktion som ”gissar” upprepade gånger. Ifall en gissning leder till att det blir omöjligt att lösa pusslet efteråt, kan vi då konstatera att den gissningen var dålig. Trotts att det finns en otroligt stor mängd alternativ att gissa på är förhoppningen att vi kan få ett snabbt program. Detta kommer vi att uppnå genom att gissa på ett smart sätt.

5. Till att börja med behöver vi ett sätt att avgöra ifall pusslet är löst eller inte. Skriv en funktion som avgör detta. Funktionen bör returnera True ifall pusslet är avklarat och False annars. Utan denna funktion skulle vår lösare inte veta när den ska sluta gissa.

6. Skriv en solve-funktion. Givet ett pussel ska denna funktion returnera True ifall det går att lösa pusslet. Ifall det går att lösa pusslet ska även pusslet uppda-

teras. Ifall pusslet inte går att lösa ska ni returnera False och pusslet ska inte ha förändrats.

Som exempel, låt säga att ni har valt att representera en Sudoku som en 9×9 matris. Innan ni anropar solve-funktionen kommer det då att finnas vissa delar av matrisen som inte är ifyllda. Ifall eran solve-funktion returnerar True bör matrisen vara helt fylld med siffror i slutet. Ifall funktionen returnerar False bör matrisen fortfarande ha ofyllda delar i slutet.

Inuti funktionen: välj den tomma ruta som är högst upp. Ifall det finns fler tomma rutor som delar plats som högsta, ta den som är längst till vänster utav dem. Loopa sedan igenom alla möjliga alternativ för den rutan och använd dem som gissningar, ett i taget. För varje gissning: fyll i rutan med eran gissning och gör sedan ett rekursivt anrop till solve. Ifall solve returnerar False, testa nästa alternativ osv. Ifall inget av alternativen fungerade, returnera False.

7. Testa att köra erat program. Vad händer? Varför? Finns det något vi kan göra bättre?

Ifall ingenting verkar hända kan det vara bra att försöka leta reda på varför det är fallet. Antingen har ni en bugg nånstans i eran kod, eller så kanske detta är förväntat.

Ifall erat program klarar av att lösa uppgiften är det värt att tänka på ifall tiden det tog känns rimlig. Testa med ett par olika pussel av olika svårighetsgrader.

8. Skriv en funktion som räknar ut ett bra ställe att gissa på. Ni får själva välja vad som menas med bra, men funktionen bör returnera den ruta ni tror är smartast att gissa på.

9. Modifiera solve-funktionen så att den gissar på eran ”smarta ruta” som ni räknade ut i förra uppgfiten.

10. Testa återigen att köra erat program. Lyckas ni lösa pusslet? Hur lång tid tar det? Ifall eran kod inte löser pusslet inom en hyffsat kort tid är det förmodligen dags att göra två saker: felsöka koden och hitta smartare rutor att gissa på.

Mätning av programmets förmåga

Nu när vi har ett program som klarar av att lösa vårt pussel är det dags att se hur mycket det klarar av. Klarar den av även de svåraste pusslena? Hur många pussel kan vi lösa inom ett kort tidsintervall?

11. Inför en räknare för antalet rekursiva anrop som görs. Detta kan vara en global variabel som uppdateras i början av varje solve-anrop. Detta kommer ge ett mått på hur bra våra gissningar är: ju färre anrop, desto bättre gissningar.

12. Kör programmet på ett par pussel och tajma hur lång tid det tar. Skriv även ut hur många anrop till solve som görs.

Extra problem

13. Försök att få ner antalet rekursiva anrop så långt som möjligt. Tänk dock på att programmet fortfarande ska vara minst lika snabbt. Det är ju inte värt att skriva en ”smartare” algoritm ifall den är långsamare.

14. Leta upp ett liknande pussel till det pussel du lyckades lösa. Till exempel om du valde sudoku kan du nu testa att lösa diagonalsudokus (vanliga sudokuregler förutom att diagonalerna även måste innehålla alla tal från 1 till 9 exakt en gång). Kopiera din lösare till en ny fil. Har du strukturerat koden på ett bra sätt behöver du endast ändra på den funktionen som kollar ifall pusslet är löst och den funktionen som letar upp var du ska gissa.

    Efter modifieringarna, testa ifall programmet klarar av den nya typen av pussel.

15. Välj en typ av pussel som inte går ut på att fylla i ett rutnät med siffror. Till exempel nonogram eller staketpussel (slitherlink på engelska). Vad behöver man nu ändra på för att lösa ett sådant pussel? Tänk lite och försök sedan att skriva en lösare på liknande sätt som tidigare.

16. Kan man använda en lösare för att skapa pussel? Försök tänka ut ett sätt att modifiera din kod så att du producerar pussel istället för att lösa pussel. Hur kan du modifiera svårighetsgraden på de pussel du skapar?

3. Computer Vision

30 juni

Idag ska vi gå igenom hur datorer (och robotar!) kan se sin omvärld, och detektera saker. Vi kommer skapa ett liknande program som fågelholken ni fick testa under skattjakten. Den använder sig av bakgrundssubtraktion med Mixture of Gaussians (MoG). Vi kommer använda oss utav OpenCV.

1. Skapa ett program som öppnar kameraströmmen och visar bilden i ett fönster. Låt Escape och tangenten Q stänga ned fönstret.

2. Skapa en MOG2 bakgrundsmodell utanför loopen och uppdatera bakgrundsmodellen med bilden, kanske med 0.0001 som learning rate. Om bilden får mycket noise och små områden eller pixlar som flimmrar så testa att blurra den innan du använder den för att uppdatera modellen, med en kernel på kanske 20x20 pixlar. Rita ut den nya förgrundsbilden i ett nytt fönster.

3. Thresholda bort alla skuggor så de blir svarta. Kvar ska du bara få det som rör sig i bilden. Kanske rita ut trösklade bilden igen i ytterligare ett nytt fönster.

4. Skapa en bild i GIMP eller annat ritprogram med mönstret som man ska matcha mot. Se till att bilden har samma upplösning som kameraströmmen. Kolla upplösningen på kameraströmmen med vanliga numpy shape. Ladda in bilden utanför while-loopen såklart! Nu ska vi ta reda på hur många pixlar som matchar med förgrunden, det kan du göra genom att ANDa mönstret med förgrunden `cv2.bitwise_and(img1, img2)`. Skapa fönster och rita ut bilderna om du har problem. Till sist ska du räkna antalet vita pixlar (eg. icke-svarta) med `cv2.countNonZero(img)`. Observera hur dumt de har namngett sina funktioner, både med camelCase och snake_case, typiskt stora opensource projekt som många committat till ...

5. Räkna ut antalet pixlar som INTE matchar mönstret. Enklast är att bara göra som ovan, fast först invertera mönstret med hjälp av `cv2.bitwise_not(img)`.

6. Räkna ut hur många procent av området som är korrekt täckt med

$$K = \frac{\text{Korrekt täckta pixlar}}{\text{Totalt antal pixlar inuti mönstret}}$$

Sedan kan vi räkna ut hur stor procent av området utanför som är täckt:

$$F = \frac{\text{Felaktigt täckta pixlar}}{\text{Total antal pixlar utanför mönstret}}$$

Du får räkna antalet pixlar utanför respektive inuti mönstret med `cv2.countNonZero(img)` som vi gjorde i uppgift 4.

Vår totala ”procent”-sats får vi genom att multiplicera ihop första förhållandet med det andra förhållandets komplement:

$$\text{tot} = K \cdot (1 - F)$$

Skriv ut den totala procentsatsen (glöm inte multiplicera med 100 för att få ett värde mellan 0 och 100 istället för 0 och 1) på bilden med:

```
font = cv2.FONT_HERSHEY_SIMPLEX
fontScale = 1
color = (200, 0, 0) # Vad för färg tror du detta är innan du testar ???
thickness = 2
img = cv2.putText(img, str(procent) + '% covered', (210,60), font, fontScale,
    color, thickness, cv2.LINE_AA)
```

7. Om procentsatsen är över ett visst värde, säg 45, så ska koden skrivas ut på bilden!

8. **Extra:** Visualisera vad algoritmen tycker att den matchar som korrekta pixlar och felaktiga pixlar genom att tona de korrekta gröna och de felaktiga röda i den ursprunliga färgbilden.

Använd den ANDade bilden mellan mönstret och förgrunden för att visualisera de korrekta pixlarna, och den ANDade bilden mellan inverterade mönstret och förgrunden för att visualisera de felaktiga pixlarna.

För att göra den ANDade bilden grön är det enklast att använda numpys inbyggda funktionalitet med villkorad indexering, dvs att indexera alla pixlar som är vita och sätta dem till grönt:

```
green = np.copy(frame)
green[(correct_image==255)] = [0,255,0]
```

Och för att blenda ihop de gröna pixlarna med den ursprungliga färgbilden kan man använda sig av:

```
blended_green = cv2.addWeighted(green, 0.3, frame, 0.7, 0)
```

Gör samma sak för att få fram de röda pixlarna och blenda in dem. Är det något som är oklart så testa rita ut bilderna i egna fönster för att se vad de innehåller.

9. **Extra:** Rita ut en kontur runt det korrekta området. Använd dig av `cv2.Canny()` för att hitta kanten av området och `cv2.findContours()` respektive `cv2.drawContours()` för att rita ut kanten. Google is your friend :)

Figur 3.1: Fredrik leker i skogen. Exempel på visualisering.

4. Numeriska metoder

2 Juli

Stjärnmärkta ($\star$) uppgifter är tänkta att vara lite mer givande, vill du inte göra alla uppgifter är dessa att rekommendera.

Rekommendation Använd dig av `numpy` för att förenkla matematiska beräkningar med arrayer för att lagra data och diverse funktioner (b.la. sin, cos och tan). För att rita funktioner med Python kan `matplotlib` användas, kort intro finns här:

`https://matplotlib.org/stable/tutorials/introductory/pyplot.html`

4.1 Numerisk ekvationslösning

Definition. *Bisektionsmetoden* är en metod för att hitta det tal x som uppfyller $f(x) = 0$ för en given kontinuerlig funktion f.

1. Betrakta ett kort intervall $[a, b]$ där f byter tecken.
2. Välj en ny punkt i mitten av intervallet $c = \frac{a+b}{2}$.
3. Om c inte är ett nollställe så har $f(c)$ samma tecken som antingen $f(a)$ eller $f(b)$. Välj ett nytt intervall $[a, c]$ eller $[c, b]$ på vilket f byter tecken.
4. Upprepa punkt 1-3 tills nollstället är inom ett tillräckligt litet intervall.

Definition. *Sekantmetoden* är en numerisk metod för att hitta det tal x som uppfyller $f(x) = 0$. Givet två tidigare approximationer, x_n och x_{n-1} så kan en förbättrad approximation x_{n+1} av x^* fås:

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

Felet för n:te iteration kan approximeras till $|x_{n+1} - x_n|$

Lös nedanstående problem med metoderna ovan.

1. Hitta en rot till funktionen med två korrekta decimaler: $f(x) = 4x^3 + 4x^2 + 6$ med hjälp av bisektionsmetoden.

2. Hitta en av lösningarna till ekvationen: $x^7 + 9x^3 + 9x - 6 = 5x + 5$ med 5 korrekta decimaler.

3. Hitta de två lösningarna som är närmast $x = 0$ till ekvationen $\arctan x = 2\sin(\pi x + 2)$ med 5 korrekta decimaler.

4. $\star$ Hitta en lösning till ekvationen $-3\ln(\sin 2x) - 2 = 0$, $x \in [-1, 1]$ med 8 korrekta decimaler. Lös först med bisektionsmetoden och sedan med sekantmetoden, jämför antalet iterationer som behöver göras!

5. ⋆ Hitta lösningen som ligger i närheten av $x = 2$ med 8 korrekta decimaler till $x^4 - 5.39x^3 + 7.2207x^2 + 3.9471x - 9.6636$. Lös först med bisektionsmetoden och sedan med sekantmetoden, jämför antalet iterationer som behöver göras!

6. Ställ upp en ekvation som låter dig approximera $\sqrt{2}$. Beräkna sedan värdet med 10 korrekta decimaler. Jämför med numpys `np`.

4.2 Integralberäkning

Definition. *Trapetsregeln* är en metod för att numeriskt approximera bestämda integraler genom att approximera värdet för delsegment som arean av en trapets.

$$\int_a^b f(x)dx \approx h \sum_{k=1}^{N} \frac{f(x_k) + f(x_{k+1})}{2} = \frac{h}{2}(f(x_1) + 2f(x_2) + 2f(x_3) + \ldots + f(x_N))$$

där $x_1 = a$, $x_N = b$ och h är steglängden.

Definition. *Monte Carlo integration* är en särskild Monte Carlo metod som numeriskt aproximerar en bestämd integral.

$$\int_a^b f(x)dx \approx \frac{(b-a)}{N} \sum_{i=1}^{N} f(x_i)$$

där x_i är uniformt distribuerade slumptal på intervallet $[a, b]$.

7. Implementera trapetsregeln som en funktion i Python.

8. Testa implementationen genom att beräkna integralen nedan och jämföra med den analytiska lösningen:

$$\int_0^{\pi} \frac{1}{1+x^2}\, dx = \arctan \pi$$

9. Implementera Monte Carlo integrationsmetoden som en funktion i Python.

10. Testa Monte Carlo metoden på integralen nedan, använd $N = 10000$. Jämför resultatet med analytiska lösningen, vad händer om du beräknar integralen igen?

$$\int_0^{\pi} x^3 \cos x\, dx = -3(\pi^2 - 4)$$

11. ⋆ Beräkna integralen nedan med hjälp av MC-metoden och med trapetsregeln. Jämför med analytiska lösningen och fundera på vilken metod som är effektivast.

$$\int_0^{\pi} \sin(x) e^x\, dx = \frac{1 + e^{\pi}}{2}$$

4.3 ODE-lösning

Sista området som för denna lektion är numerisk lösning av ordinära differentialekvationer. Ordinära differentialekvationer av första ordningen kan skrivas på formen:

$$\dot{y} = f(x, y)$$

Där f är någon funktion av variabeln x och funktionen själv $y(x)$ och $\dot{y}$ är första derivatan av y. Ekvationen betraktas som löst när om vi hittat funktionerna som uppfyller ekvationen.

Vid numerisk lösning så kan vi inte hitta ett uttryck för funktionen, men vi kan approximera funktionens värde. Detta räcker ofta för att lösa verkliga problem.

Metoderna vi kommer studera här behandlar en särskild typ av problem som är väldigt vanliga inom fysik och teknik, begynnelsevärdesproblem (BVP). Vi har då en funktion eller ett system av funktioner i en variabel, fysikaliskt är det ofta tid. Tillsammans med begynnelsevärden, vad systemets tillstånd är vid ett specifikt värde på variabeln.

Idén bakom metoderna här är att använda oss av linjarisering för att approximera nästa funktionens värde vid nästa tidpunkt baserat på föregående tidpunkt. Genom att göra detta om och om igen så kan vi få fram precis så många punkter vi vill.

Definition. *Euler framåt (RK1)* är en metod för att numeriskt lösa begynnelsevärdes problem beskrivna av första ordningens ǧekvationer. Definitionen lyder:

$$y^{n+1} = y_n + h \cdot y_t^n$$

Där y^n är lösningens värde vid det n:te tidssteget, och y_t^n är derivatans värde vid det n:te tidssteget, h är steglängden.

12. Lös följande BVP med RK1 och steglängden $h = 0.4$:

$$\dot{y} = y$$

$$y(0) = 2,\ 0 \leq t \leq 4$$

13. Lös följande BVP med RK1:

$$\dot{y} = -y + \sin t$$

$$y(0) = 2,\ 0 \leq t \leq 4$$

Jämför med den analytiska lösningen: $y = \frac{1}{2}(5e^{-x} + \sin x - \cos x)$. Testa dig fram till lämplig steglängd.

4.3.1 Lösning av ODE system

ODE system följer väldigt naturligt från ODE:er, istället för att endast ha en ekvation så har vi flera. Dessa ekvationer kan också vara kopplade, dvs. bero på

varandra. Ett system av två ekvationer kommer se ut något i stil med detta:

$$\begin{cases} \dot{x} = f(t,x,y) \\ \dot{y} = g(t,x,y) \end{cases}$$

När vi hanterar mer än en ekvation så kan vi lätt hamna i situationer där RK1 inte räcker till. För många system av ODE:er så växer RK1 till oändligheten trots att analytiska lösningen inte gör det. Då behöver vi en annan metod som exempelvis RK4.

Definition. *Runge-Kutta 4 (RK4)* är en metod för att numeriskt lösa första ordningens differentialekvationer. För ett BVP:

$$y' = f(t,y),\ y(0) = a$$

Kan RK4 skrivas som:

$$\begin{aligned} w_1 &= f(t_n, y_n) \\ w_2 &= f(t_n + \frac{1}{2}h, y^n + h\frac{w_1}{2}) \\ w_3 &= f(t_n + \frac{1}{2}h, y^n + h\frac{w_2}{2}) \\ w_4 &= f(t_n + h, y^n + hw_3) \\ y^{n+1} &= y^n + \frac{h}{6}(w_1 + 2w_2 + 2w_3 + w_4) \end{aligned}$$

RK4 kan ses som aningen komplicerad, men den är bara en vidareutveckling av RK1. Istället för att endast linjarisera i föregående punkt för att få nästa så approximerar derivatan ett halvt tidssteg framåt (w_2, w_3) och ett tidsteg framåt (w_4). Sedan tar vi ett medelvärde av alla dessa approximationer av derivatan och gör ett tidssteg framåt.

Lösningsmetoden för system är annars precis samma som för en ekvation, vi behöver bara se till att vår implementation utför beräkningarna för alla ekvationer i systemet.

Lös nedanstående problem om RK4 och ODE system:

14. ⋆ Undersök nedanstående BVP med RK4 och RK1 med h = 1:

$$\dot{y} = 0.1y$$

$$y(0) = 1,\ 0 \leq t \leq 10$$

Jämför felet i de sista tre tidsstegen med den analytiska lösningen: $y = e^{0.1t}$, vilken metod verkar ge bäst resultat?

15. ⋆ Undersök nedanstående BVP med RK1 och RK4 med h = 0.5:

$$\dot{y} = -5y$$

$$y(0) = 2,\ 0 \leq t \leq 30$$

Den analytiska lösningen är $y = 2 \cdot e^{-x}$. Vilken metod ger bäst resultat, ger båda korrekta lösningar?

16. Lös med RK4:

$$\begin{cases} \dot{x} = -x \\ \dot{y} = x \end{cases}$$

För $0 \leq t \leq 10$, $x(0) = 1$, $y(0) = 2$

17. Lös med RK4:

$$\begin{cases} \dot{x} = -y \\ \dot{y} = x \end{cases}$$

För $0 \leq t \leq 10$, $x(0) = 1$, $y(0) = 0$

4.3.2 Omskrivning till första ordnings system

Både RK1 och RK4 bygger på en approximation av första derivatan, för att kunna applicera dessa högre ordnings ODE:er så behöver de skrivas om till system av första ordnings ODE:er. Detta görs genom substitution, exempelvis för en 2:a grads ekvation som nedan:

$$\ddot{y} = -\dot{y} - y$$

Substitution: $u = \dot{y}$

$$\Rightarrow \ddot{y} = \dot{u} = -u - y$$

Vi får ett system: $\begin{cases} \dot{u} = -u - y \\ \dot{y} = u \end{cases}$

Detta system kan hanteras precis som vanliga ODE system och vi kan då få fram y.

Lös nu problemen nedan:

18. Skriv om ODE:en nedan till ett system av första ordnings ODE:er och lös med RK4:

$$\ddot{y} = -\dot{y} - y$$

$$\dot{y}(0) = y(0) = 1,\ 0 \leq t \leq 8$$

Använd steglängden $h = 0.5$

19. Skriv om ODE-systemet nedan till ett system av första ordnings ODE:er och lös det med RK4:

$$\begin{cases} \ddot{x} = -4x \\ \ddot{y} = x - \dot{y} - y \end{cases}$$

Lös för $t \in [0,\ 30]$:

$$\dot{x}(0) = 0,\ x(0) = 1,\ \dot{y}(0) = 1,\ y(0) = 0$$

4.3.3 Utmanande problem (för den som gillar fysik)

20. Vikt som hänger i fjäder med dämpning. Längd fjäder: $l = 4$ meter, fjäderkonstant $k = 4$ N/m massa vikt: $m = 1$ kg, dämpningskoefficienten γ och tyngdaccelerationen är $g = 9.81$ m/s^2. Begynnelsevilkoret, tillståndet vid t=0 är $x = 1.5$ m och $\dot{x} = 0$ m/s.
Dämpningskoefficient: beskriver en kraft som är proportionell till hastigheten men motsatt riktad. Dvs. $F_d = -\gamma\dot{x}$
Upg a. Använd RK4 för att lösa problemet med $\gamma = 0.5$ Ns/m mellan tiden 0 s och 20 s.
Upg b. Använd RK4 för att lösa problemet med $\gamma = 0$ Ns/m mellan tiden 0 s och 30 s.
Upg c. Vad händer efter om lösningen till b. beräknas för mycket lång tid (ex. $t = 2000$, steglängd $h =$)? Jämför med analytiska lösningen nedan:

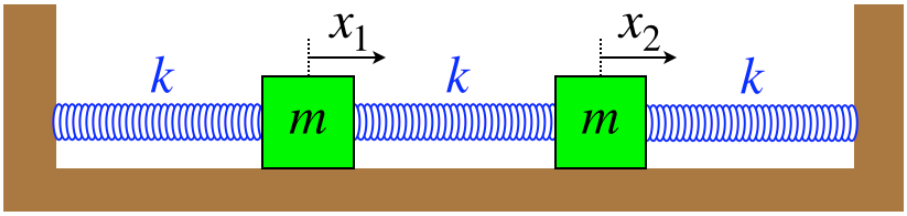
$$x(t) = 1.5\sin 2t + 1.5\cos 2t$$

Ledning 1: Vi behöver inte ta med tyngdkraften då den endast leder till en förskjutning av jämviktsläget och därför är ointressant. Kraften från fjädern kan beräknas via Hookes lag: $F_{fj} = -kx$, där x är förskjutningen från jämvikt. Dämpningen kan på motsvarande sätt ses som en kraft verkandes på vikten $F_d = -\gamma\dot{x}$. Använd sedan newtons andra lag.
Ledning 2: Newtons andra lag säger att

$$ma = -\gamma\dot{x} - kx,\ a = \ddot{x}$$

$$\Leftrightarrow \ddot{x} = m^{-1}(-\gamma\dot{x} - kx)$$

21. Studera två massor fästa i varandra med en fjäder och fästa i väggen. Likt vad som syns på videon här: `https://www.youtube.com/watch?v=UpseJGf_l2w`. Vi ska försöka undersöka detta system med våra numeriska metoder.



Säg att fjäderkonstanterna är $k_{1,2} = 2$ N/m för de yttre fjädrarna och $k_3 = 1$ N/m för mittenfjädern. Låt massorna vara $m = 1$ kg. Använd $x_1(0) = 0$, $\dot{x}_1(0) = 0.5$, $x_2(0) = 0.5$, $\dot{x} = 0$ (m resp. m/s). Lös för $0 \leq t \leq 30$ sek med RK4.
Ledning 1: Krafterna på det vänstra blocket kan beräknas till:

$$F_1 = -k_1 x_1 + k_3(x_2 - x_1)$$

Krafterna på det högra kan beräknas till:

$$F_2 = -k_2 x_2 + k_3(x_1 - x_2)$$

Ledning 2: Newtons andra lag ger då detta ODE system ($m = 1$):

$$\begin{cases} \ddot{x}_1 = -x_1(k_1 + k_3) + k_3 x_2 \\ \ddot{x}_2 = +k_3 x_1 - x_2(k - 2 + k_3) \end{cases}$$

Undersök: Vad händer vid olika kombinationer av begynnelsevärden för blocken?

5. Reinforcement learning

3 juli

Definition. ***Reinforcement learning*** handlar om att träna en AI att bete sig på ett visst sätt eller att utföra en viss uppgift genom att ge den någon slags digital belöning när den gör något bra, och att ge ett digitalt straff när den gör något dåligt. ***Q-learning*** är en specifik algoritm som kan uppdatera Q-funktionen efter varje steg som AI:n tar.

Definition. En ***policy*** talar om vilken ***action*** (=rörelse) en AI ska utföra i varje ***state*** (=tillstånd). Värdefunktionen V ger ett värde på hur bra varje tillstånd s är att vara i. Q-funktionen ger för varje action i ett tillstånd de förväntade framtida belöningarna. V-funktionen i ett tillstånd är maximum av alla Q-värden i tillståndet, och den optimala policyn är att genomföra den action med högst Q-värde.

Definition. Den Q-funktion som ni ska använda ser ut så här:

$$Q(s_k, a) = r(s_k, a) + \gamma V^*(s_{k+1})$$

Vidare så gäller i denna definition att:

$$V^*(s_k) = \max_a Q(s_k, a)$$

För att AI:n ska kunna lära sig Q-funktionen behövs en algoritm. Den ni ska använda ser ut så här:

$$\hat{Q}(s_k, a_j) \leftarrow (1-\eta)\hat{Q}(s_k, a_j) + \eta(r + \gamma \hat{V}(s_{k+1}))$$

Eta är här en parameter som talar om hur mycket vi vill ta hänsyn till det precis gjorda steget för att uppdatera vår kunskap om Q. Om eta är lika med 0 kommer Q aldrig förändras, om eta är lika med 1 så kommer Q-värdet att uppdateras helt i varje nytt steg i iterationen.
Här är r belöningen att utföra alternativ j, i state k, och γ är en discount factor < 1 som säger att vi värdesätter belöning nu direkt mer än förväntat belöning längre fram.

Så här kan pseudo-kod för Q-learning se ut:

Initialize a look-up table for *Q(s,a)* with random values.
for each episode
  Init a start state *s*
  repeat for each step *k* in the episode
    Choose an action a_j (see next slides)
    Take action a_j and observe reward *r* and next state s_{k+1}
    Update estimated Q-function:

$$\hat{Q}(s_k, a_j) \leftarrow (1-\eta)\hat{Q}(s_k, a_j) + \eta\left(r + \gamma \max_a \hat{Q}(s_{k+1}, a)\right)$$

  end
end

En sista sak att tänka på är hur vi ska välja steg att utföra. När vi börjar så har vi troligtvis inga bra Q-värden, därför är det inte alls säkert att vi borde följa det som är vår uppskattning av den optimala V-funktionen. Inför istället en parameter ε som säger att vi med denna sannolikhet tar ett helt slumpmässigt steg, och med sannolikhet $1 - \varepsilon$ följer det som vår Q-funktion säger.

1. Läs igenom instruktionerna i Instructions.pdf
2. Titta på koden i QLearning, utils och world-filerna! Vad gör den? Vilka funktioner finns? Vad saknas som ni behöver implementera själva?
3. Följ resten av instruktionerna i den andra filen. En bra början är att generera de olika världarna och titta på hur de ser ut, samt att köra runt roboten manuelt med hjälp av explicita anrop.
4. Skriv kod som tränar upp en AI så att den lär sig att navigera de olika banorna! Ett tips är att visualisera vad som händer under träningen!
5. Skrv kod som testar den färdiga AI:n och visar hur den rör sig i världen!

Häftet innehåller material från Mattekollo 2022: alla matematikkurserna samt materialet för alla programmeringsgrupperna.

Tack till alla deltagare och ledare!