

Perspektiverende Datalogi

Klassiske Algoritmer

Gerth Stølting Brodal

Ugens Program

Mandag

10.15-12.00 Introduktion til Algoritmik
Gerth Stølting Brodal

Tirsdag

9.15-12.00 Øvelser - Open Learning Center

12.15-13.00 Opgave 11: Længste voksende delsekvens
Gerth Stølting Brodal

13.15-16.00 Øvelser - Open Learning Center

Onsdag

14.15-16.00 Historisk perspektiv
Erik Meineche Schmidt

Algoritmer

Algoritme Klart beskrevet metode til løsning af en opgave

Eksempler

2 dl havregryn
4 dl vand
Hæld alt i gryde.
Kog 3 min.
Smag til med salt.

Madopskrift

50-35-30 g Tvinni
to-trådet grøn
Pinde nr. 3

Slå 38-28-20 m op,
strik 4-3-3 p glatstr,
start med r p. Lav
raglan-indtag 2 r 2
dr r sm.

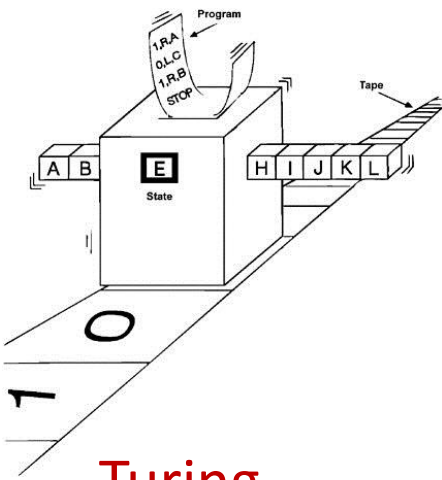
Strikkeopskrift

```
int i,k;  
for (i=0;i<N;i++){  
    C[A[i]]++;  
    k = k+i;  
}
```

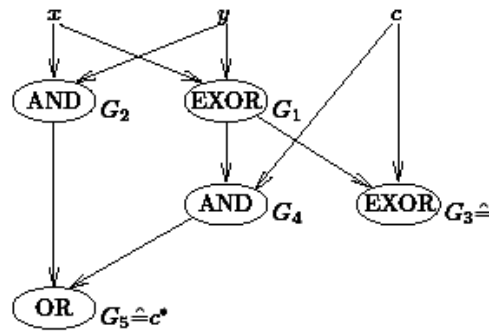
Computerprogram

Beregningsmodeller

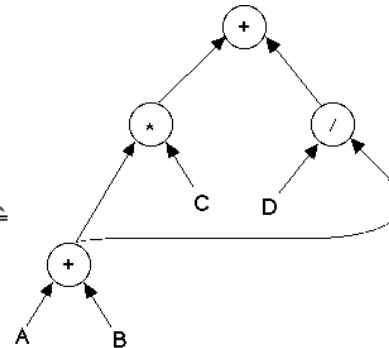
Formel model: Beskriver præcis hvad en algoritme kan gøre, præcis definition af resourceforbrug



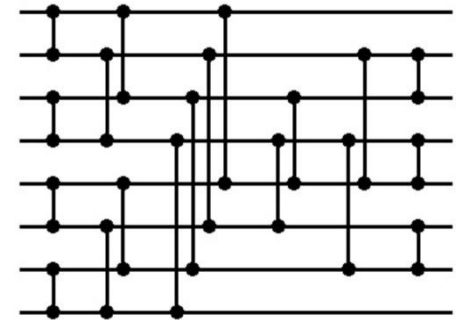
Turing
maskine



Boolske
netværk



Aritmetiske
netværk



Sorterings
netværk
(Øvelse 6)

Algoritmik

= designe og analysere algoritmer

Kvalitet af algoritme:

- **Korrekt**, d.v.s. løser bevisligt problemet
- Effektiv - lavt **ressourceforbrug**, f.eks.
 - Tid
 - Plads
- Nem at programmere
- Problem-specifikke egenskaber



Søgning i Sorteret Liste

3

7

9

11

13

27

33

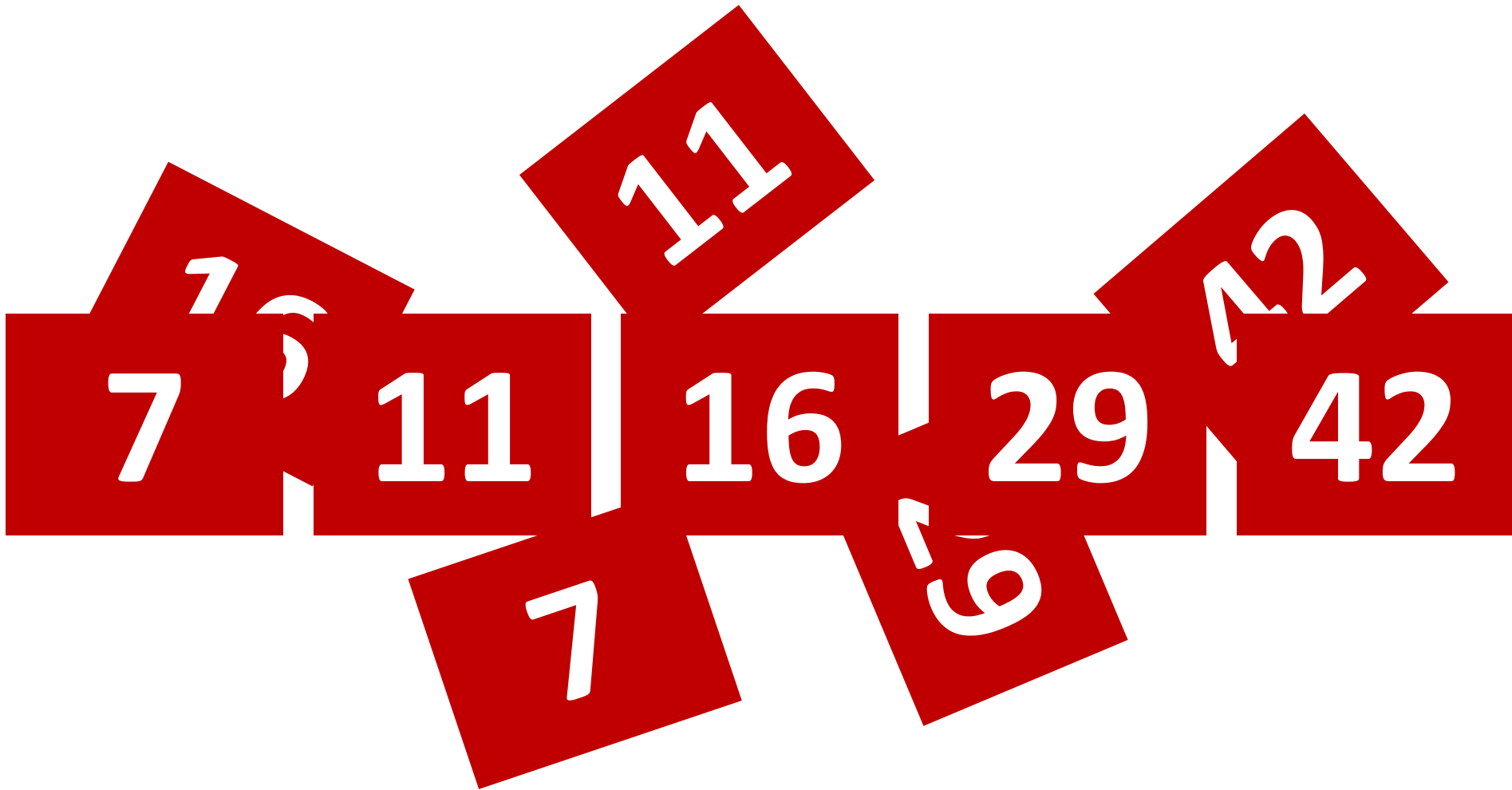
37

42

89

$1 + \lceil \log_2 n \rceil$ sammenligninger

Sortering



(Øvelser 1-10)

Begreber

- Analyse af algoritme
 - **assymptotisk tid**
 - **øvre grænse**
 - **worst-case**
 - **best-case**
- Analyse af problem
 - **nedre grænse** (alle algoritmer tager lang tid)
 - **adversary/modspiller** (strategi for at finde et dårligt input for en given algoritme)
 - **tælleargumenter** (tæl "mængden af information" en algoritme opnå) (**øvelser 9**)
- Modeller
 - **sorteringsnetværk** (**øvelser 6**)
 - **beslutningstræer** (**øvelser 9**)

Matematik repetition

$$1 + 2 + 3 + \dots + N = \frac{N(N+1)}{2}$$

$$1 + 2 + 4 + \dots + 2^{k-1} + 2^k = 2^{k+1} - 1$$

x	1	2	4	8	...	64	...	80	...	128	...
$\log_2(x)$	0	1	2	3		6		6.3219		7	

\uparrow
 $2^6 = 64$

$$y = \log_2(x) \Leftrightarrow x = 2^y$$

$$\log_2(x \cdot y) = \log_2(x) + \log_2(y)$$

NB: Ascii notation ofte $2^3=2^{\wedge}3$

$$\log(x) = \log_2(x)$$

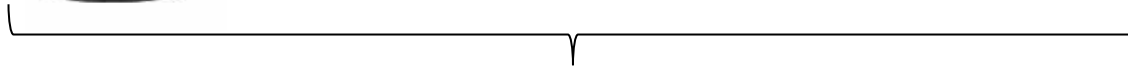
$$\log(x) \neq \log_{10}(x)$$

$$\log(x) \neq \log_e(x) = \ln(x)$$

NB: I datalogi...

Husk

Hver læsegruppe skal tirsdag medbringe: en **saks**, to-tre **ure** med sekundvisere, **skriveredskaber** og lidt kladdepapir, evt. en **lommeregner** (gerne grafisk)



Indhold

- **Eksempler på beregningsproblemer**
- Algoritmer og deres analyse
 - Korrekthed af algoritmer
 - Ressourceforbrug for algoritmer
- Komplexitet af beregningsproblemer

Beregningsproblemer

- Sortering
- Søgning
- Grafer
- Streng
- Geometri
- Numeriske beregninger
- Kombinatorisk optimering
- ...



Sortering

- **Problem** Stil en mængde elementer i orden

110 755 766 51 652 28 729 713 681 407



28 51 110 407 652 681 713 729 755 766

- Data er meget bekvemmere hvis de er sorterede. Specielt er det nemmere at lede i dem (ordbøger, telefonbøger, eksamensopslag, ...)
- Brugt som rutine i mange andre algoritmer
- Meget velstuderet problem
- Mange algoritmer (**øvelser 1-10** + QuickSort + ...)

Søgning

- **Problem:** Gem data så de kan findes igen effektivt

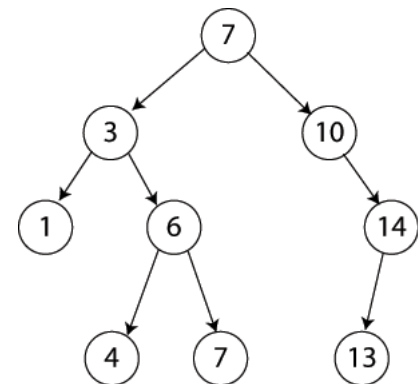
Find(x)

Insert(x)

Delete(x)

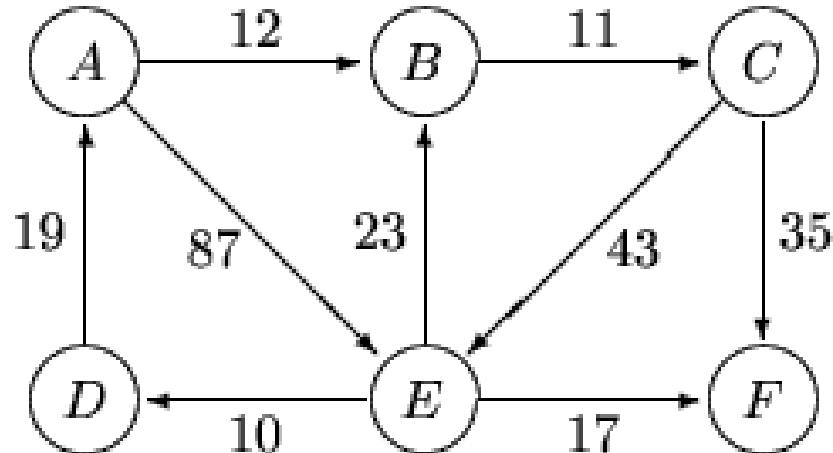
Successor(x), RangeSearch(x_1, x_2), ...

- Et andet meget fundamentalt problem (jvf. databaser)
- Brugt som rutine i mange andre algoritmer
- Meget velstuderet, mange algoritmer
- To grundgrupper: søgetræer og hashing



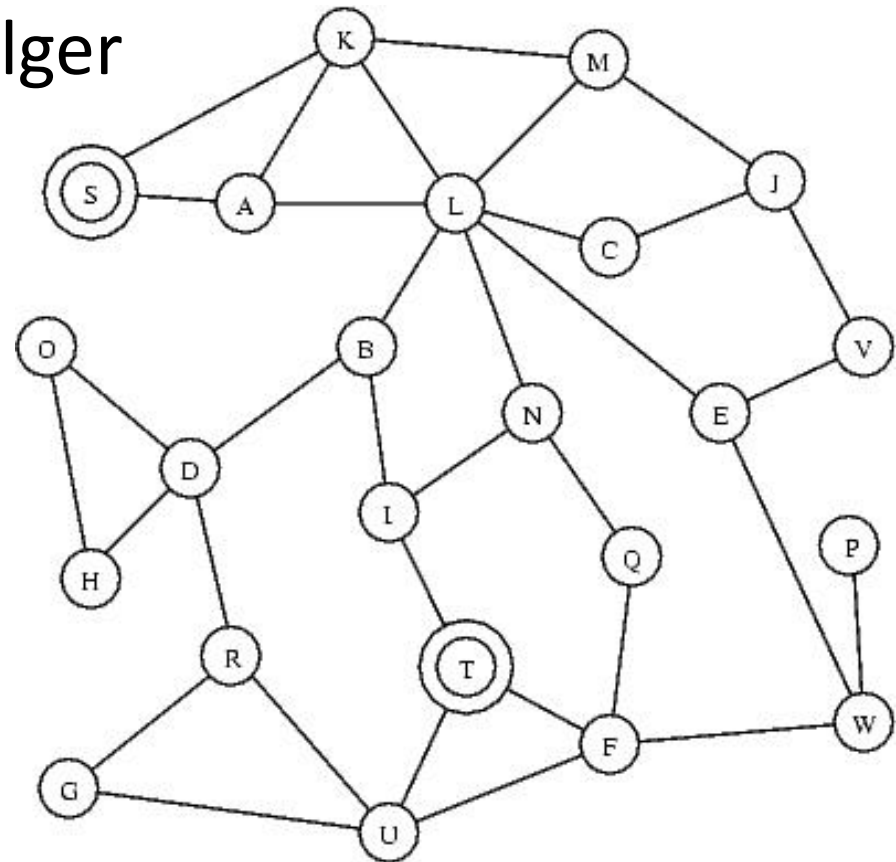
Grafer

- **Knuder** (punkter) og **kanter** (streger mellem punkter)
- Ekstra struktur: **orientering** af kanter, **vægte** på kanter
- En *meget* anvendt model:
Flyruter, veje, el/vand/computer netværk, bekendtskaber og andre relationer, weblinks, ...



Problemer på grafer

- Løb grafen igennem (besøge alle knuder)
- Sammenhæng, k-sammenhæng
- (Mindste) udspændende træ
- Hamilton tur, rejsende sælger
- Korteste veje
- Euler tur
- Graffarvning
- Klike
- ...



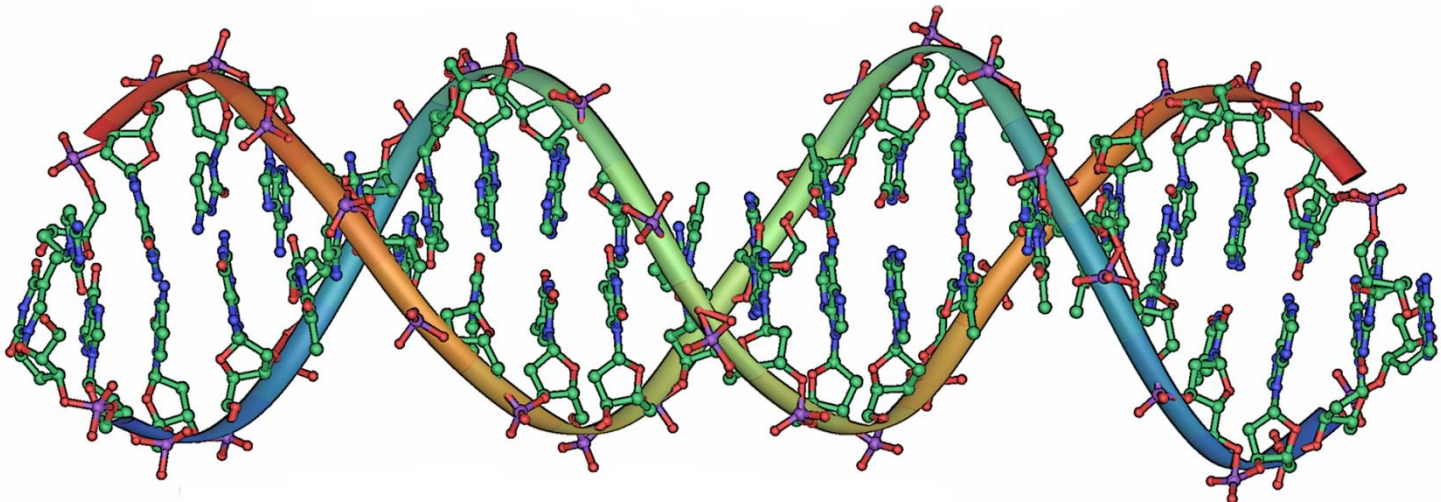
Streng

- **Alfabet** = mængde af tegn som kan bruges
- **Streng** = sekvens af tegn fra alfabetet
- Eksempler:

“To be or not to be”

10001100110001110

ACCCATTCCGTAA



Problemer på Streng

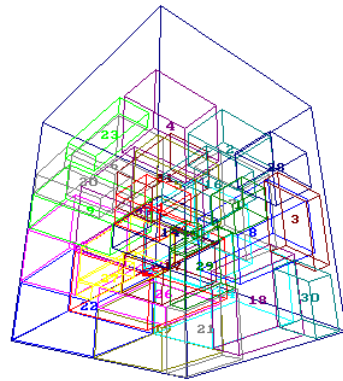
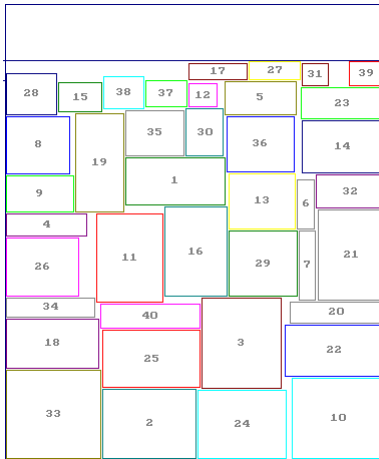
- Mønster genkendelse
- Regulære udtryk
- Afstandsmål (Hamming, edit)
- Længste fælles delstreng
- Længste fælles delsekvens
- ...

ACGTTAC**GTAAACTACTAGTACACACACTACCCAT**

AC**AGTGT**TAA**AGTAAACTAAACTAGTACACACGTACCCAT**

Kombinatorisk Optimering

- Spande pakning (1D, 2D, 3D)



- Rygsæk
- Delmængde sum
- Job-skemaplanlægning
- Flight/crew-planlægning
- ...

Ferie

Flyrejser

Rundrejser på egen hånd

- Mexico**: 12 dages blikvinkel på Mexico City 19.900,-
- Cuba**: 12 dages rundrejse i det vestlige Cuba 18.290,-
- Sydafrika**: 8 dages blikvinkel på Johannesburg og Cape Town 15.755,-
- USA**: "Sunny Southern highlights" i Californien 17.000,-

Storby-ferie på egen hånd

- Irland**: 7.999,-
- Tog i Schweiz**: 11.499,-
- Berlin**: 2.599,-
- Piemonte/Barolo**: 2.999,-
- Prag**: 2.999,-
- Rom**: 599,-

Kvalitätsrejser med dansk rejseleder

- Provence**: 6.999,-
- Atlantis Rejser**: 6.999,-

Rødehavet bedst & billigst

- Rødehavet All inclusive**: 3.499,-
- Golf og koreveje**: 3.499,-
- Krydstogt på Nilen**: 6.999,-

AGUST UDSELG!

OKER HETTER MED NADAT	RESTPLADSER til 2020
ALPINE ALPES - 2020 100,-	12/21 November 2020 200,-
ALPINE ALPES - 2020 100,-	13/22 November 2020 200,-
ALPINE ALPES - 2020 100,-	14/23 November 2020 200,-
ALPINE ALPES - 2020 100,-	15/24 November 2020 200,-
ALPINE ALPES - 2020 100,-	16/25 November 2020 200,-
ALPINE ALPES - 2020 100,-	17/26 November 2020 200,-
ALPINE ALPES - 2020 100,-	18/27 November 2020 200,-
ALPINE ALPES - 2020 100,-	19/28 November 2020 200,-
ALPINE ALPES - 2020 100,-	20/29 November 2020 200,-
ALPINE ALPES - 2020 100,-	21/30 November 2020 200,-
ALPINE ALPES - 2020 100,-	22/01 December 2020 200,-
ALPINE ALPES - 2020 100,-	23/02 December 2020 200,-
ALPINE ALPES - 2020 100,-	24/03 December 2020 200,-
ALPINE ALPES - 2020 100,-	25/04 December 2020 200,-
ALPINE ALPES - 2020 100,-	26/05 December 2020 200,-
ALPINE ALPES - 2020 100,-	27/06 December 2020 200,-
ALPINE ALPES - 2020 100,-	28/07 December 2020 200,-
ALPINE ALPES - 2020 100,-	29/08 December 2020 200,-
ALPINE ALPES - 2020 100,-	30/09 December 2020 200,-
ALPINE ALPES - 2020 100,-	31/10 December 2020 200,-

Bestil på tlf. 70 11 40 11

Jordans højdepunkter

Victors Formor

ved at vide?

CHARTERFERIE FOR HELE FAMILIEN

SUN TOURS

PARIS

CYPERN

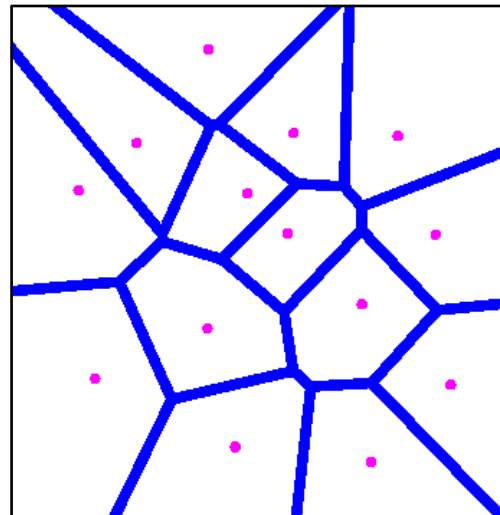
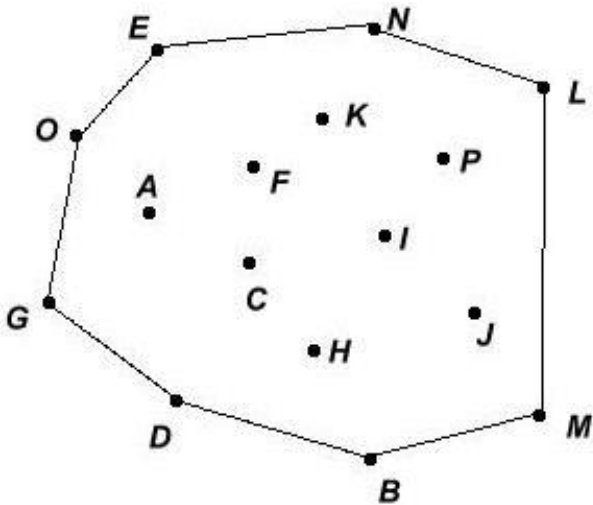
AGATA

OPLEV TOSCANA!

ved at vide?

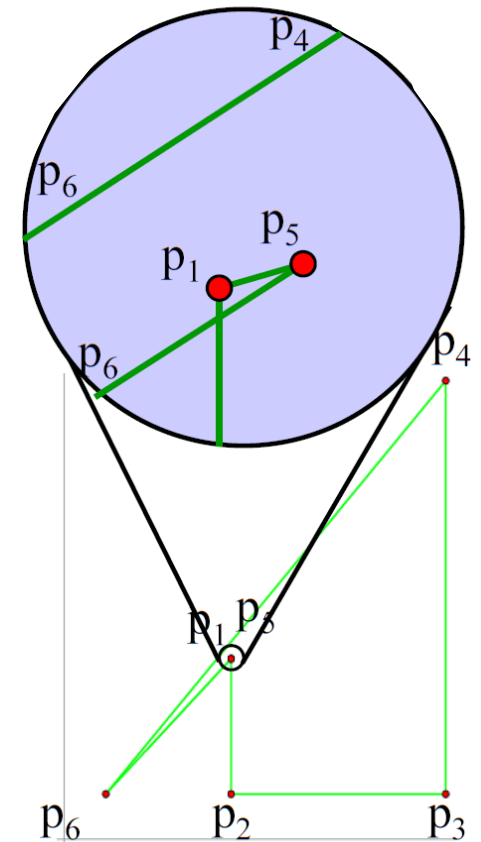
Geometri

- Konvekse hylster
- Nærmeste naboer
- Krydsninger af ortogonale linjesegmenter
- 2D område søgninger
- ...



Numeriske Beregninger

- Polynomieevaluering
- Matrixmultiplikation
- Løsning af ligningssystemer
- Løsning af differentialligninger
- ...



$$(6/7-1)*7+1 = -4.44089209850063e-16 \text{ ?}$$

$$0.1+0.2 \neq 0.3$$

Indhold

- Eksempler på beregningsproblemer
- Algoritmer og deres analyse
 - **Korrektthed af algoritmer**
 - Ressourceforbrug for algoritmer
- Komplexitet af beregningsproblemer

Invarianter

Udsagn / som gælder efter alle skridt i algoritmen

Vælges så:

- Man kan vise at / gælder ved **starten**
- Man kan vise at hvis / gælder før et **skridt**, så gælder det efter
- Man kan vise af / samt omstændigheder ved algoritmens **afslutning** implicerer det ønskede slutresultat

Eksempler: Binær søgning, RadixSort (**øvelse 7**), ...

3

7

9

11

13

27

33

37

42

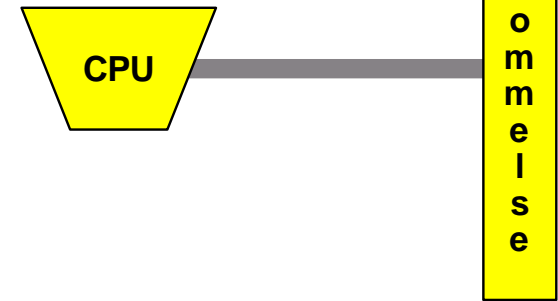
89

Indhold

- Eksempler på beregningsproblemer
- Algoritmer og deres analyse
 - Korrekthed af algoritmer
 - **Ressourceforbrug for algoritmer**
- Komplexitet af beregningsproblemer

Ressourceforbrug, målestok

- RAM-modellen
- Tidsmåling vs. analyse
- Voksehastighed, asymptotisk notation
- Worst case, best case, average case
- ...



Først vælge (eller designe) algoritme efter forskelle i **asymptotisk ressourceforbrug**

Ved lighed, vælg **dernæst** efter **konstanter**
(tidsmåling nu relevant)

Indhold

- Eksempler på beregningsproblemer
- Algoritmer og deres analyse
 - Korrekthed af algoritmer
 - Ressourceforbrug for algoritmer
- **Kompleksitet af beregningsproblemer**

Kompleksitet

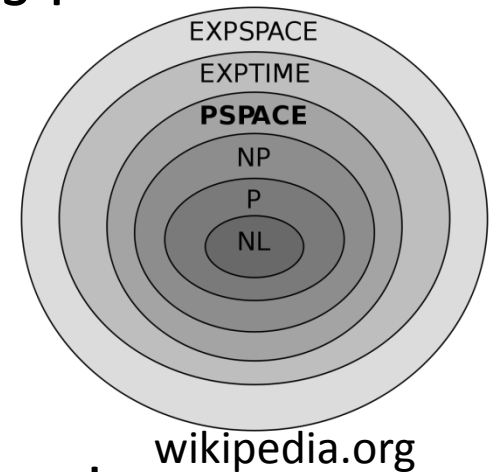
Kompleksitetsteori
= studere problemers iboende sværhedsgrad

Kompleksitetsklasser:

Klasse(**X**,**Y**) = De problemer, som kan løses i model **X** med ressourceforbrug **Y**.

Mål

Øvre grænser (dvs. algoritmer) og **nedre grænser** (dvs. beviser for at **ingen** algoritme i model **X** kan løse problemet med ressourceforbrug mindre end **Y**).



Nedre Grænser

- Beviser for at **ingen** algoritme (blandt en stor klasse af algoritmer for en given beregningsmodel) kan løse problemet bedre end angivet
- Eksempler: Søgning og Sortering

Øvre og nedre grænser ens



problemets kompleksitet kendt

$$P \subseteq EXP$$

Meget grov inddeling af algoritmer i gode og dårlige:

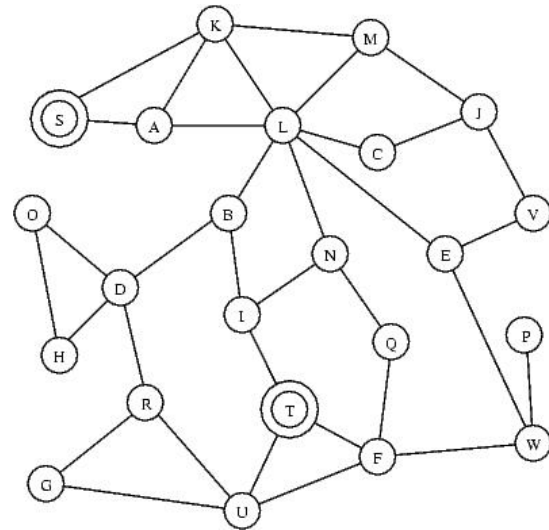
- P = problemer med polynomiell tids algoritme
- v.s.
- EXP = problemer med eksponentiel tids algoritme

Eksempel: sortering vs. brute-force løsning af puslespil
(øvelse 17)

NP

- NP = ja/nej-problemer, hvor en ja-**løsning kan kontrolleres** (men ikke nødvendigvis findes) i polynomiel tid

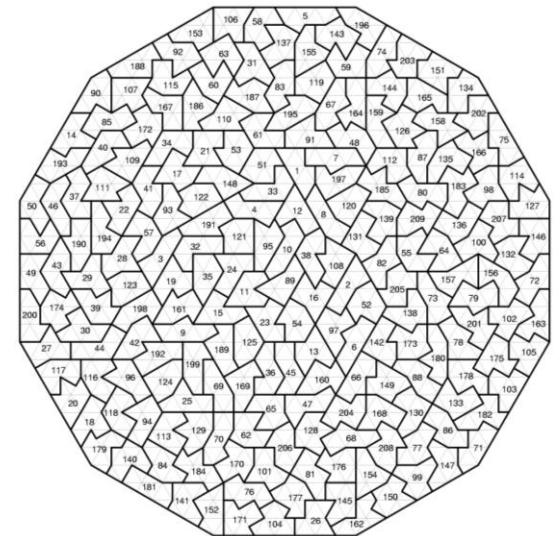
- Eksempel: Hamilton tur



- Det er nemt at se at $P \subseteq NP \subseteq EXP$
- Formodning: $P \subset NP$

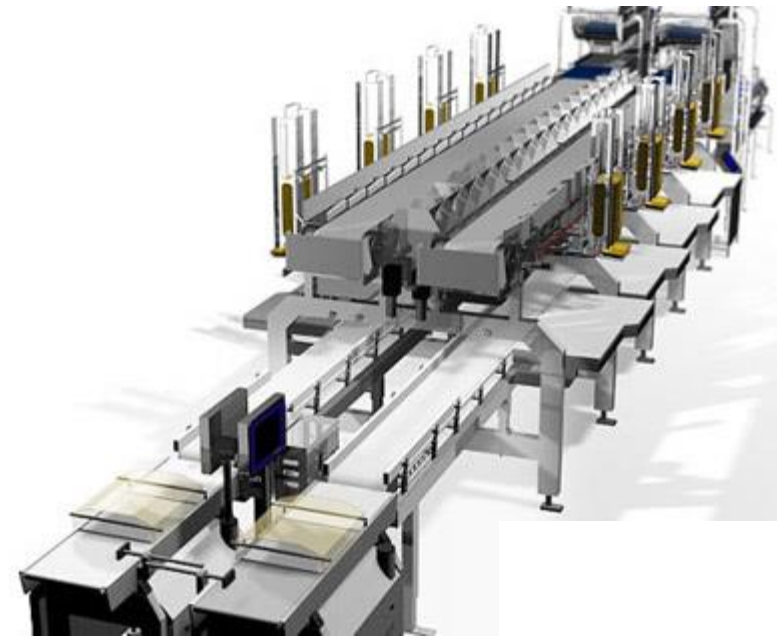
Hvis ingen polynomiel algoritme...

- Heuristisk søgning
- Algoritmer for specielle instanser (jvf. “The Eternity Puzzle”)
- Approximationsalgoritmer



Flere modeller og cost-funktioner

- Online algoritmer
- Randomiserede algoritmer
- Parallelisme
- Hukommelseshierarkier
- ...

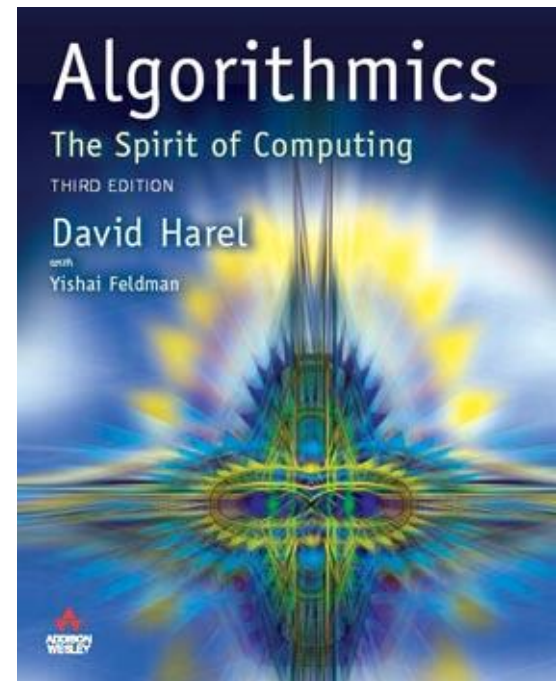


Algoritmik

- Designe algoritmer
- Analyse algoritmer
- Analyse problemer

David Harel

*“Algorithmics is more than a branch of computer science. It is the **core of computer science**, and, in all fairness, can be said to be relevant to most of science, business, and technology.”*



Øvelse 11 (Længste voksende delsekvens)

Betragt følgende liste af tal.

30 83 73 80 59 63 41 78 68 82 53 31 22 74 6 36 99 57 43 60

Øvelsen er at slette så få af disse tal som muligt, så de resterende tal står i voksende orden. Hvis for eksempel alt på nær de første to tal slettes, har man en voksende følge tilbage af længde 2 (**30 83**). Ved at slette alt på nær det første, tredje, ottende og tiende opnår man det samme, men har slettet færre tal og har en voksende følge tilbage af længde 4 (**30 73 78 82**).

Spørgsmål

Hvor lang er den længste voksende følge man kan opnå ?