Perspektiverende Datalogikursus

Perspektiverende Datalogikursus

Uge 1 - Algoritmer og kompleksitet

Gerth Stølting Brodal

2. september 2005

Afleveringsopgaver...

```
                   _
     /\ ..  // \\
    /  \||  |   |
   / [] \|  \\_//
  /      \
 /________\
  |      |
  | []   |
  |   ._.|
  |   | ||
  |   | ||
----------------
```

Øvelse 10

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...

Indhold

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

Beregningsproblemer

- Sortering
- Søgning
- Grafer
- Strenge
- Kombinatorisk optimering
- Geometri
- Numeriske beregninger

  ⋮

Sortering

Problem: Stil en mængde elementer i orden.

110 755 766 51 652 28 729 713 681 407

$$\Downarrow$$

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 (jvf. øvelserne + QuickSort + ShellSort + . . .).

Søgning

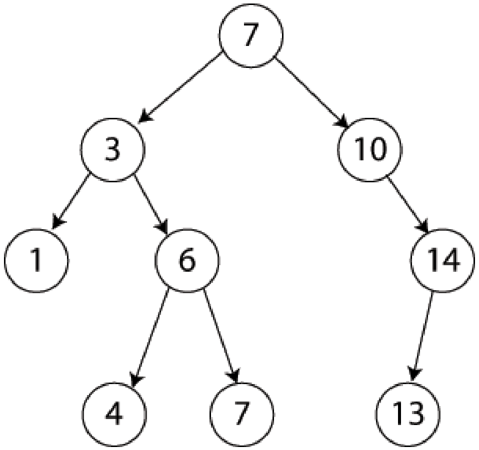
Problem: Gem data så de kan findes igen effektivt.

- 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.



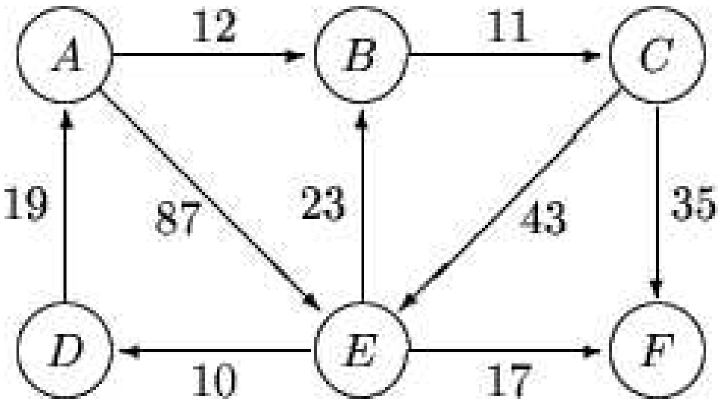
Grafer

Knuder (punkter) og kanter (streger mellem punkter).

Ekstra struktur: orientering af kanter, vægte på kanter.

En *meget* anvendelig model:

> Flyruter, veje, el/vand/computer netværk, bekendtskaber og andre relationer,. . .



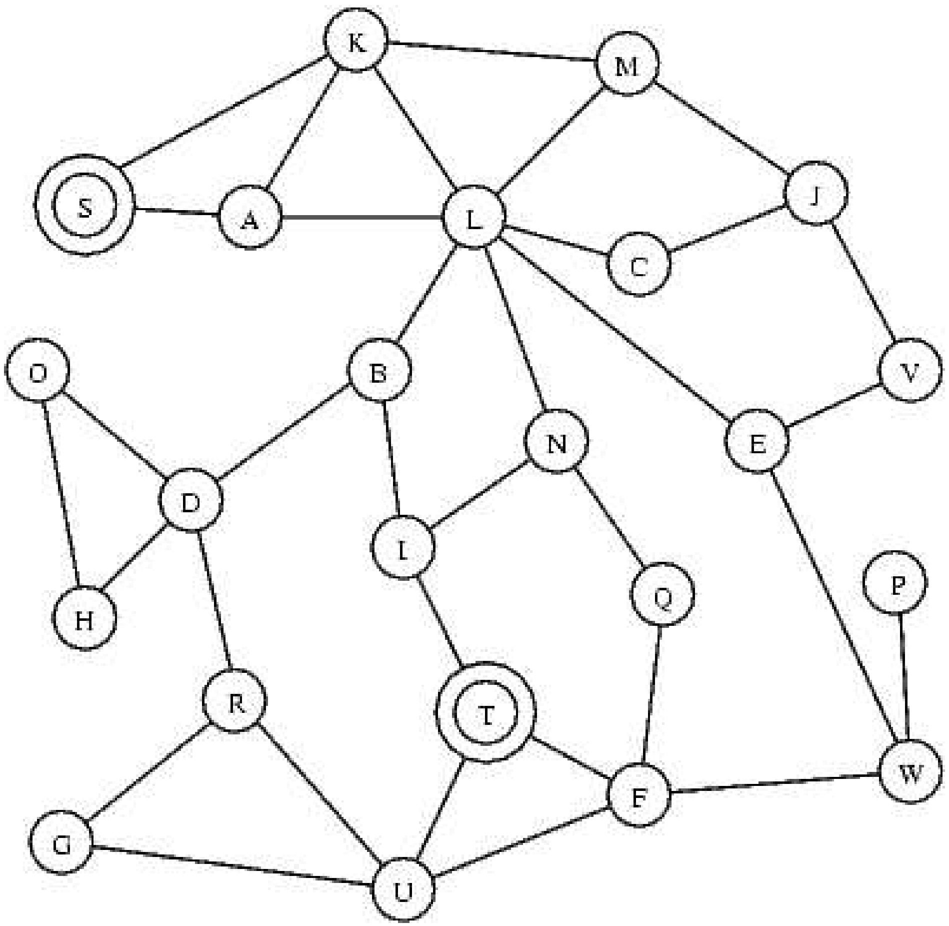
Perspektivenrende Datalogikursus

Perspektiverende Datalogikursus

Problemer på grafer

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

$\vdots$



Strenge

Alfabet = mængde af tegn som kan bruges

Streng = sekvens af tegn fra alfabetet

Eksempler:

"To be or not to be"

ACCCATTCCGTAA

10001100110001110

Perspektiverende Datalogikursus

Problemer på strenge

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

⋮

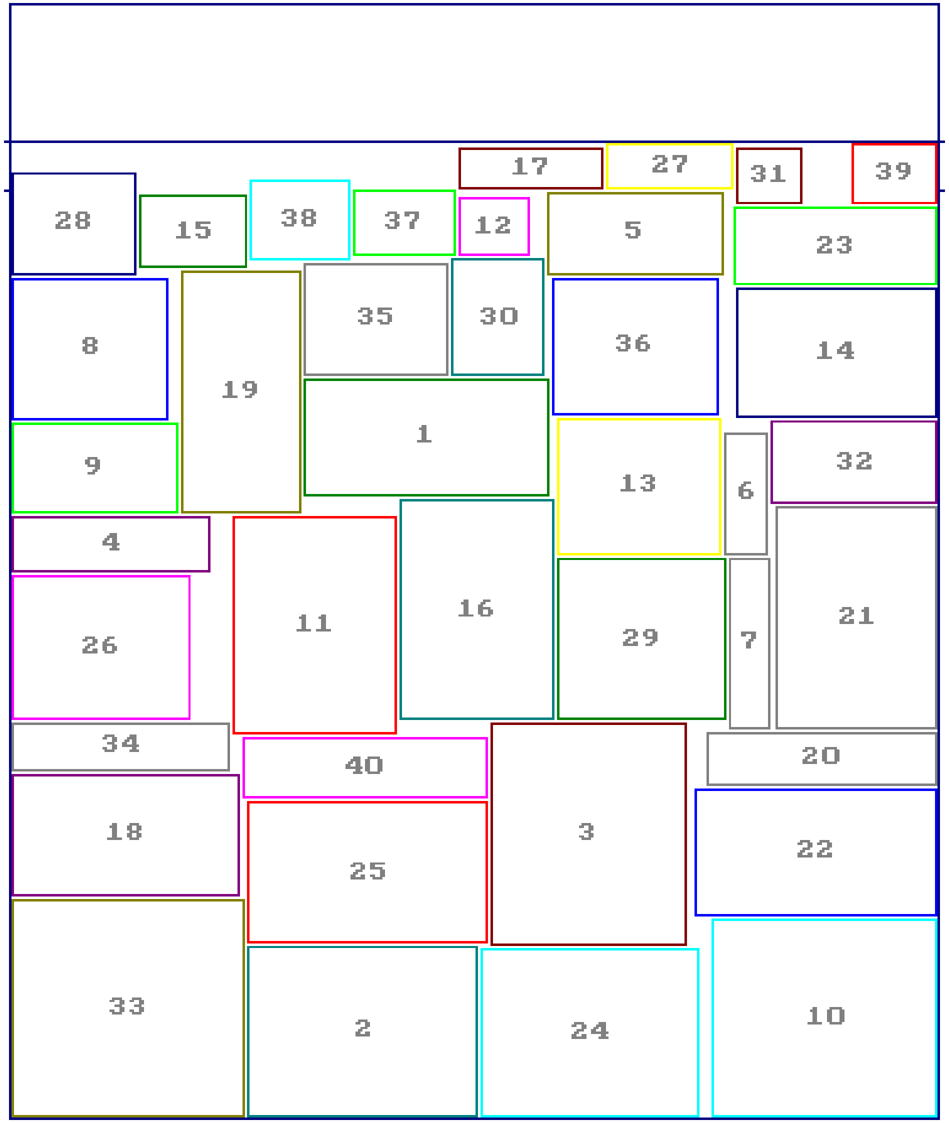
ACGTTACGTAAACTACTAGTACACACACTACCCAT

ACAGTGTTAAGTAAACTAAACTAGTACACACGTACCCAT

Kombinatorisk optimering

- Bin Packing (1D, 2D, 3D)
- Knapsack
- Subset Sum
- Job Scheduling
- Crew assignment

⋮



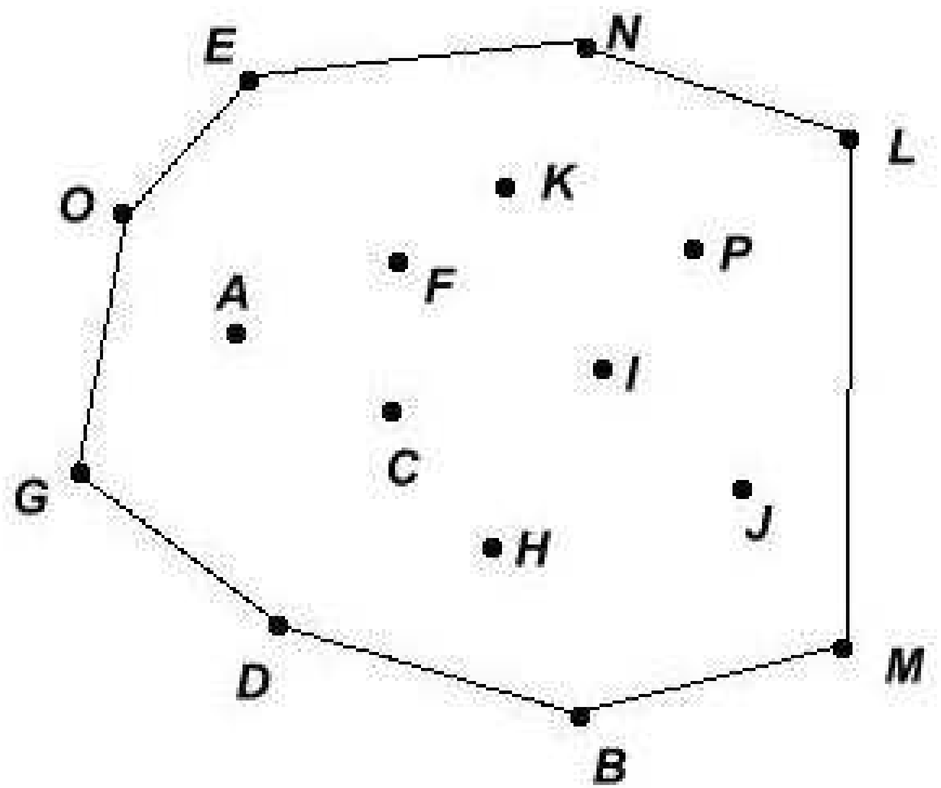
Perspektiverende Datalogikursus

Perspektiverende Datalogikursus

Geometri

- Convex Hull
- Nearest Neighbor
- Ortogonal Line Segment Intersection
- 2D Range Search

⋮



Numeriske beregninger

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

⋮

(6/7-1)*7+1 = -4.44089209850063e-16 ?

Perspektiverende Datalogikursus

Indhold

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

Invarianter

Udsagn I som gælder efter alle skridt i algoritmen.

Vælges så:

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

Eksempler: RadixSort, binær søgning.

Perspektiverende Datalogikursus

Indhold

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

Ressourceforbrug, målestok

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

Først vælge (eller udvikle) 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**

Nedre grænser

Beviser for at ingen algoritme (blandt en stor klasse af algoritmer for en given beregningsmodel) kan løse problemet bedre end angivet.

Eksempler:

- Sortering
- Søgning

Øvre og nedre grænser ens

$\Downarrow$

problemets kompleksitet kendt.

$\mathbf{P} \subseteq \mathbf{EXP}$

Meget grov inddeling af algoritmer i gode og dårlige:

P = problemer med polynomiel tids algoritme

v.s.

EXP = problemer med eksponentiel tids algoritme

Eksempel: sortering vs. brute-force løsning af puslespil

NP

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

Eksempel: Hamilton tur.

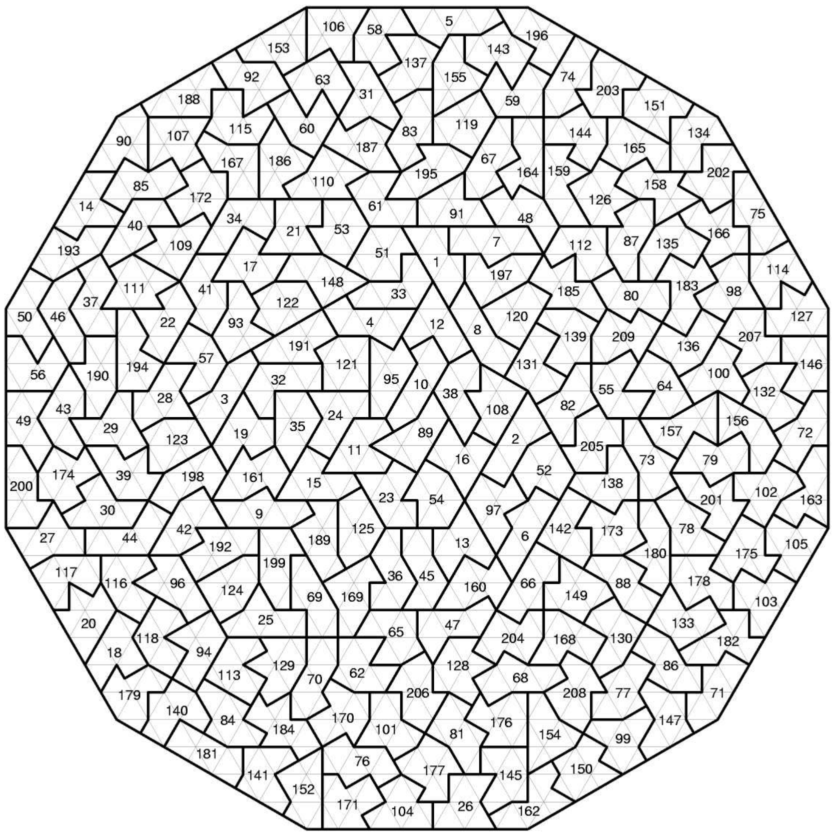
Det er nemt at se at P $\subseteq$ NP $\subseteq$ EXP.

Formodning: P $\subsetneq$ NP

Perspektiverende Datalogikursus

Hvis ingen polynomiel algoritme..

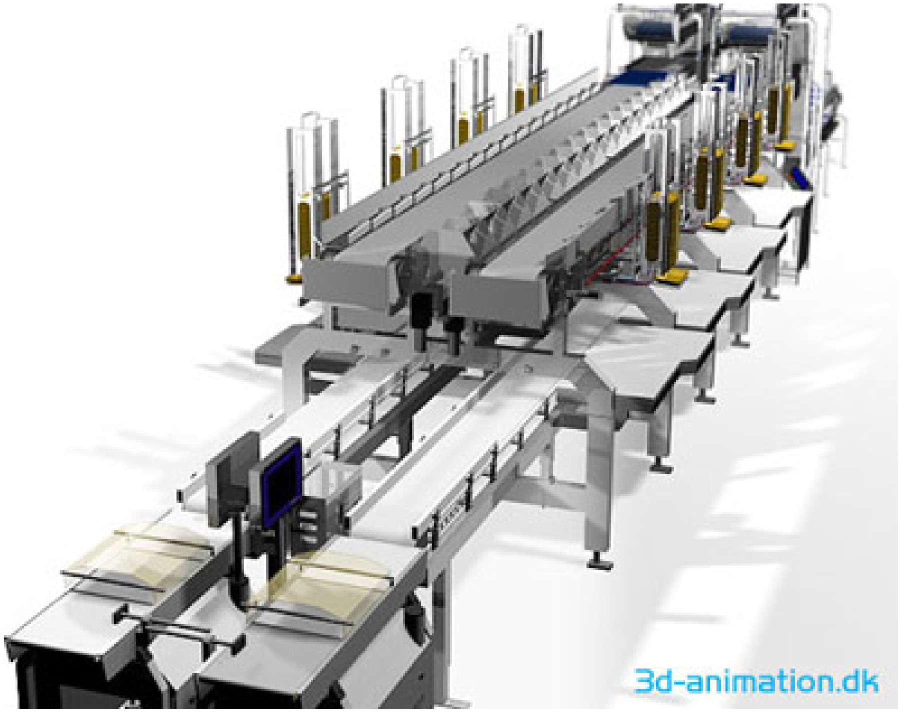
- Heuristisk søgning
- Algoritmer for specielle instanser
  (jvf. “The Eternity Puzzle”).
- Approximationsalgoritmer



Flere modeller og cost-funktioner

- Online algoritmer
- Randomiserede algoritmer
- Parallelisme
- Hukommelseshierarkier

⋮



Algoritmer og kompleksitet

- Udvikle algoritmer
- Analysere algoritmer
- Analysere 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."*