

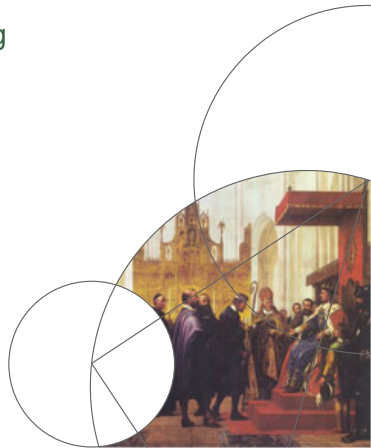


Det Naturvidenskabelige Fakultet

Studiepraktik 2018

Øvelser i Algoritmer og problemløsning

Benjamin Rotendahl



① Algoritme vs Algoritme

② Algoritme design

③ Øvelser



Program for idag

Fra nu til 11.50

- Øvelser i algoritmer



Program for idag

Fra nu til 11.50

- Øvelser i algoritmer
- Sjove gåder



Program for idag

Fra nu til 11.50

- Øvelser i algoritmer
- Sjove gåder
- Opsamling og spørgsmål



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .

Hold A

- Har en computer

Hold B



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .

Hold A

- Har en computer
- Bruger algoritmen
Merge Sort

Hold B



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .

Hold A

- Har en computer
- Bruger algoritmen *Merge Sort*
- De kan sortere 10 millioner tal på under 20 minutter

Hold B



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .

Hold A

- Har en computer
- Bruger algoritmen *Merge Sort*
- De kan sortere 10 millioner tal på under 20 minutter
- De kan sortere 100 millioner tal på 4 timer.

Hold B



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .

Hold A

- Har en computer
- Bruger algoritmen *Merge Sort*
- De kan sortere 10 millioner tal på under 20 minutter
- De kan sortere 100 millioner tal på 4 timer.

Hold B

- Har en computer der er 1000 gange hurtigere end hold A



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .

Hold A

- Har en computer
- Bruger algoritmen *Merge Sort*
- De kan sortere 10 millioner tal på under 20 minutter
- De kan sortere 100 millioner tal på 4 timer.

Hold B

- Har en computer der er 1000 gange hurtigere end hold A
- Bruger algoritmen *Insertion Sort*



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .

Hold A

- Har en computer
- Bruger algoritmen *Merge Sort*
- De kan sortere 10 millioner tal på under 20 minutter
- De kan sortere 100 millioner tal på 4 timer.

Hold B

- Har en computer der er 1000 gange hurtigere end hold A
- Bruger algoritmen *Insertion Sort*
- De kan sortere 10 millioner tal på 5 timer.



Valget af algoritme

Soterings algoritmer

Givet en liste af n tal ønsker vi at returnere en sorteret liste af længde n .

Hold A

- Har en computer
- Bruger algoritmen *Merge Sort*
- De kan sortere 10 millioner tal på under 20 minutter
- De kan sortere 100 millioner tal på 4 timer.

Hold B

- Har en computer der er 1000 gange hurtigere end hold A
- Bruger algoritmen *Insertion Sort*
- De kan sortere 10 millioner tal på 5 timer.
- De kan sortere 100 millioner tal på **23 dage!**



- Køretid på $O(n \lg n)$

Sammenligning af Algoritmer

Vi bruger begrebet Køretid for at beskrive hvordan tiden en algoritme bruger stiger med input.



Sammenligning af Algoritmer

Vi bruger begrebet Køretid for at beskrive hvordan tiden en algoritme bruger stiger med input.

Definition på køretid

En øvregrænse for den tid der bliver brugt på at løse et problem af størrelse n . Skrives som

$$O(n), O(n^2), O(n \lg n), O(n!), O\left(\frac{a}{b}\right)$$



Sammenligning af Algoritmer

Vi bruger begrebet Køretid for at beskrive hvordan tiden en algoritme bruger stiger med input.

Definition på køretid

En øvregrænse for den tid der bliver brugt på at løse et problem af størrelse n . Skrives som

$$O(n), O(n^2), O(n \lg n), O(n!), O\left(\frac{a}{b}\right)$$

Algoritme for minimums funktionen

Givet en liste $X = [x_1, x_2, \dots, x_n]$ ønsker vi at returnere det mindste tal i listen. Hvad er algoritmen og hvad er køretiden?



Minimums algoritme

Eksempel

Algorithm 1

Input: En liste $X = [x_1, x_2, \dots, x_n]$

Output: Det mindste tal i listen.

$\text{min} = x_1$

for x_i in X **do**

if $x_i < \text{min}$ **then**

$\text{min} = x_i$

end if

end for



Minimums algoritme

Eksempel

Algorithm 2

Input: En liste $X = [x_1, x_2, \dots, x_n]$

Output: Det mindste tal i listen.

$\text{min} = x_1$

for x_i in X **do**

if $x_i < \text{min}$ **then**

$\text{min} = x_i$

end if

end for

Analyse af algoritmen

Køretid?



Minimums algoritme

Eksempel

Algorithm 3

Input: En liste $X = [x_1, x_2, \dots, x_n]$

Output: Det mindste tal i listen.

$\text{min} = x_1$

for x_i in X **do**

if $x_i < \text{min}$ **then**

$\text{min} = x_i$

end if

end for

Analyse af algoritmen

Køretid? $O(n)$



Minimums algoritme

Eksempel

Algorithm 4

Input: En liste $X = [x_1, x_2, \dots, x_n]$

Output: Det mindste tal i listen.

$\text{min} = x_1$

for x_i in X **do**

if $x_i < \text{min}$ **then**

$\text{min} = x_i$

end if

end for

Analyse af algoritmen

Køretid? $O(n)$

Er den optimal?



Minimums algoritme

Eksempel

Algorithm 5

Input: En liste $X = [x_1, x_2, \dots, x_n]$

Output: Det mindste tal i listen.

$\text{min} = x_1$

for x_i in X **do**

if $x_i < \text{min}$ **then**

$\text{min} = x_i$

end if

end for

Analyse af algoritmen

Køretid? $O(n)$

Er den optimal? **Jeps!**



Hvordan finder man på en algoritme

Algoritme for algoritmer

- 1 Beskriv problemet med egne ord.



Hvordan finder man på en algoritme

Algoritme for algoritmer

- 1 Beskriv problemet med egne ord.
- 2 Del problemet op i mindre dele.



Hvordan finder man på en algoritme

Algoritme for algoritmer

- 1 Beskriv problemet med egne ord.
- 2 Del problemet op i mindre dele.
- 3 Definer input



Hvordan finder man på en algoritme

Algoritme for algoritmer

- 1 Beskriv problemet med egne ord.
- 2 Del problemet op i mindre dele.
- 3 Definer input
- 4 Definer output



Hvordan finder man på en algoritme

Algoritme for algoritmer

- 1 Beskriv problemet med egne ord.
- 2 Del problemet op i mindre dele.
- 3 Definer input
- 4 Definer output
- 5 Beskriv trin for at gå fra input til output



Øvelser

Algoritme for øvelserne

- 1 Der præsenteres et problem med eksempler.



Øvelser

Algoritme for øvelserne

- 1 Der præsenteres et problem med eksempler.
- 2 I finder på en algoritme for problemet (Arbejd gerne sammen)



Øvelser

Algoritme for øvelserne

- 1 Der præsenteres et problem med eksempler.
- 2 I finder på en algoritme for problemet (Arbejd gerne sammen)
- 3 Vi løser den sammen på tavlen.



Øvelser

Algoritme for øvelserne

- 1 Der præsenteres et problem med eksempler.
- 2 I finder på en algoritme for problemet (Arbejd gerne sammen)
- 3 Vi løser den sammen på tavlen.
- 4 Plus gåder!



Søgning

Mål

Givet en sorteret liste og et element, bestem om element er i listen, ved at kigge på så få elementer som muligt.



Søgning

Mål

Givet en sorteret liste og et element, bestem om element er i listen, ved at kigge på så få elementer som muligt.

Eksempel

Lad en liste være givet ved $[2, 4, 5, 7, 8, 11, 25]$, hvor vi ønsker at finde ud af om elementet 11 er listen. Svaret skulle gerne være ja. (det første element har indeks 0).



Sortering

Mål

Sorter en givet usorteret liste.



Sortering

Mål

Sorter en givet usorteret liste.

Eksempel

Lad en liste være givet ved $[7, 4, 5, 12, 1]$, denne vil vi gerne sortere! Den sorterede liste skulle gerne være $[1, 4, 5, 7, 12]$.



Nøgle gåde

Hvordan holder jeg min garagedør lukket?

I har en gargeport med en IR modtager der kan modtage et signal og en nøglering der kan sende et signal.



Nøgle gåde

Hvordan holder jeg min garagedør lukket?

I har en gargeport med en IR modtager der kan modtage et signal og en nøglering der kan sende et signal.

Mål

Hvordan kan vi gøre den sikker? Hvad skal "Computeren" i nøgleringen kunne?



Nøgle gåde

Hvordan holder jeg min garagedør lukket?

I har en gargeport med en IR modtager der kan modtage et signal og en nøglering der kan sende et signal.

Mål

Hvordan kan vi gøre den sikker? Hvad skal "Computeren" i nøgleringen kunne?

Eksempel

Send en kode til IR modtageren?



Overlappende under problemer

Eksempel

Det n 'te *Fibonacci* tal er defineret som summen af de to forgående.

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \dots$$


Overlappende under problemer

Eksempel

Det n 'te *Fibonacci* tal er defineret som summen af de to forgående.

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \dots$$

Hvordan bestemmer vi dem?



Sommer i Sunny beach

Beskrivelse

Det er sommer i *Sunny Beach* og en kæde af barer langs kysten har et problem,



Sommer i Sunny beach

Beskrivelse

Det er sommer i *Sunny Beach* og en kæde af barer langs kysten har et problem, de kan ikke bestille flere øl!



Sommer i Sunny beach

Beskrivelse

Det er sommer i *Sunny Beach* og en kæde af barer langs kysten har et problem, de kan ikke bestille flere øl!

Hver bar langs strandvejen har b_i øl tilbage, der er ikke nogen der ved hvilken bar der kan sælge flest øl på en given aften.



Sommer i Sunny beach

Beskrivelse

Det er sommer i *Sunny Beach* og en kæde af barer langs kysten har et problem, de kan ikke bestille flere øl!

Hver bar langs strandvejen har b_i øl tilbage, der er ikke nogen der ved hvilken bar der kan sælge flest øl på en given aften. Det er besluttet at alle barer skal have lige mange øl.



Sommer i Sunny beach

Beskrivelse

Det er sommer i *Sunny Beach* og en kæde af barer langs kysten har et problem, de kan ikke bestille flere øl!

Hver bar langs strandvejen har b_i øl tilbage, der er ikke nogen der ved hvilken bar der kan sælge flest øl på en given aften. Det er besluttet at alle barer skal have lige mange øl.

De har en lastbil hvor der kan være en uendelig mængde af øl,



Sommer i Sunny beach

Beskrivelse

Det er sommer i *Sunny Beach* og en kæde af barer langs kysten har et problem, de kan ikke bestille flere øl!

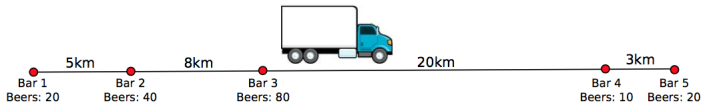
Hver bar langs strandvejen har b_i øl tilbage, der er ikke nogen der ved hvilken bar der kan sælge flest øl på en given aften. Det er besluttet at alle barer skal have lige mange øl.

De har en lastbil hvor der kan være en uendelig mængde af øl, dog er vognførerne på denne lastbil glade for øl. Hver gang der er kørt en kilometer så drikkes der to øl.



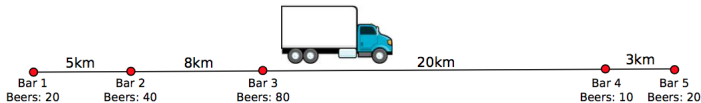
Eksempel

Figur



Eksempel

Figur



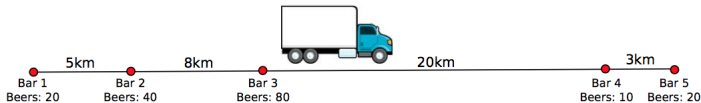
Hvad i får som input

- En liste over hvor mange øl der er på hver bar b_1, b_2, \dots
- En liste over afstanden mellem dem a_1, a_2, \dots



Eksempel

Figur



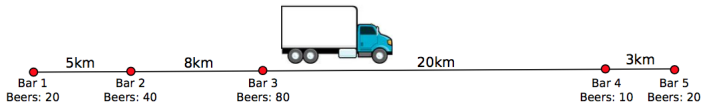
Hvad i får som input

- En liste over hvor mange øl der er på hver bar b_1, b_2, \dots
- En liste over afstanden mellem dem a_1, a_2, \dots
- Det antal øl de gerne vil have på hver bar



Eksempel

Figur



Hvad i får som input

- En liste over hvor mange øl der er på hver bar b_1, b_2, \dots
- En liste over afstanden mellem dem a_1, a_2, \dots
- Det antal øl de gerne vil have på hver bar

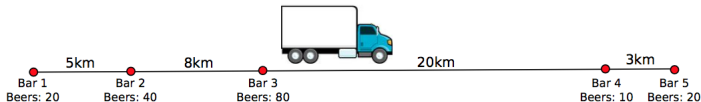
Hvad i skal svare

- ja, eller nej.



Eksempel

Figur



Hvad i får som input

- En liste over hvor mange øl der er på hver bar b_1, b_2, \dots
- En liste over afstanden mellem dem a_1, a_2, \dots
- Det antal øl de gerne vil have på hver bar

Hvad i skal svare

- ja, eller nej.
- Svar også på hvordan man kan finde det optimale



DNS system

Hvad er DNS

Forklaring på tavlen

Mål

Tænk over hvordan vi kan lave et DNS system



DNS system

Hvad er DNS

Forklaring på tavlen

Mål

Tænk over hvordan vi kan lave et DNS system

Eksempel

Snak lidt om hvordan det kunne gøre smart



Zig-Zag sekvens

Beskrivelse

Ejeren af et supermarked har fået en ny stor ølhylde, nu vil han gerne gøre den pæn. Han har placeret sine øl i alfabetisk orden, men højden på flaskerne afviger. Han er nu interesseret i at finde den længste sekvens af øl der skifter mellem lav og høj en såkaldt Zig-Zag sekvens. Så $(15, 10, 17)$ er sådan en sekvens mens $(1, 2, 3)$ ikke er. Vi skal lave en algoritme der bestemmer sådan en sekvens.

Input: n øl flaskers højde n_1, n_2, \dots

Output: Den længst mulige Zig-Zag sekvens.

Eksempel

- Input : (6, 1, 7, 7, 2, 4, 7)
- Output : (6, 1, 7, 2, 7)

Figur



I må gerne fjerne flasker men ikke bytte rundt på dem, beskriv en metode til at bestemme den her sekvens og tænk over hvor effektivt den er.



Primtal

Mål

Skriv en algoritme der finder det n 'te primtal — prøv at gøre den så hurtig som mulig.



Primtal

Mål

Skriv en algoritme der finder det n 'te primtal — prøv at gøre den så hurtig som mulig.

Eksempel

Find det femte primtal. Algoritmen skal gerne returnere 11.

