

Algoritmer og Datastrukturer

Invarianter

et værktøj til analyse og design af algoritmer

Note brugt indtil foråret 2017 i kurset
Algoritmer og Datastrukturer 1 til at
dække emnet Invarianter

Department of Computer Science
Aarhus University

Algorithms and Data Structures
Transition systems

Mikkel Nygaard.
Erik Meineche Schmidt
February 2014

[Transition Systems](#),
Mikkel Nygaard Hansen og
Erik Meineche Schmidt.
DAIMI-FN-64, Februar 2014.



invariant adjektiv

BØJNING - , -e

UDTALE ['envai,an'd]

OPRINDELSE første led latin *in-* i betydningen 'ikke-, u-'

Betydninger

uforanderlig; konstant – bl.a. inden for matematik

GRAMMATIK almindelig som substantiv fælleskøn

i ← 0

x ← 100

while i ≤ 10 **do**

x ← x + 7

i ← i + 1

Hvilken value har x når programmet stopper ?

```
i ← 0  
x ← 100  
while i ≤ 10 do  
    x ← x + 7  
    i ← i + 1
```

- a) 100
- b) 107
- c) 110
- d) 111
- e) 170
- f) 177

Løkke-invariant = udsagn

Eksempler på I

- $i \geq 0$
- $x \geq 100$
- $i \leq x$

{ I }

$i \leftarrow 0$

$x \leftarrow 100$

while $i \leq 10$ **do**

$x \leftarrow x + 7$

$i \leftarrow i + 1$

$$i=0 \wedge x=100 \rightarrow x = 100+7*i \wedge i \leq 11$$

$$x = 100 + 7 * i \wedge i \leq 11 \\ \wedge x \text{ og } i \text{ er heltal}$$

$$x=100+7*i \wedge i \leq 11 \wedge i \leq 10 \\ \wedge x'=x+7 \wedge i'=i+1$$

$$\neg(i \leq 10) \wedge (x = 100+7*i \wedge i \leq 11) \rightarrow i = 11 \wedge x = 177$$

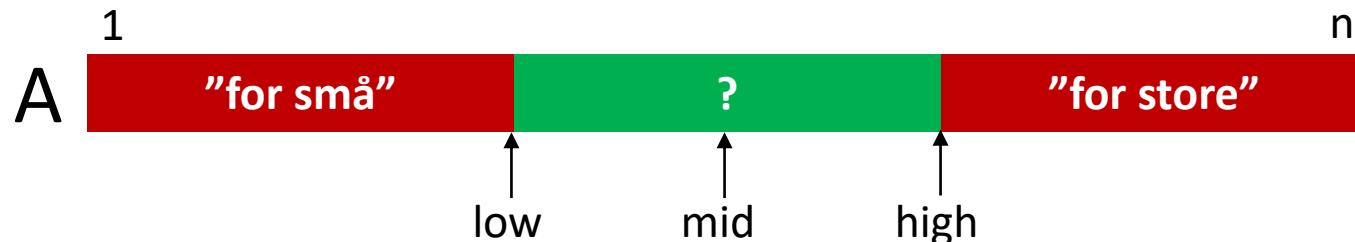
En invariant I skal opfylde

- 1) Når løkken nås første gang, så er I opfyldt
- 2) Hvis I er opfyldt før løkken udføres, så er I opfyldt efter en udførsel af løkken

I gælder automatisk når vi kommer ud af løkken

Udnyt I og at løkke-betingelsen er forkert når vi er færdig til at drage en konklusion

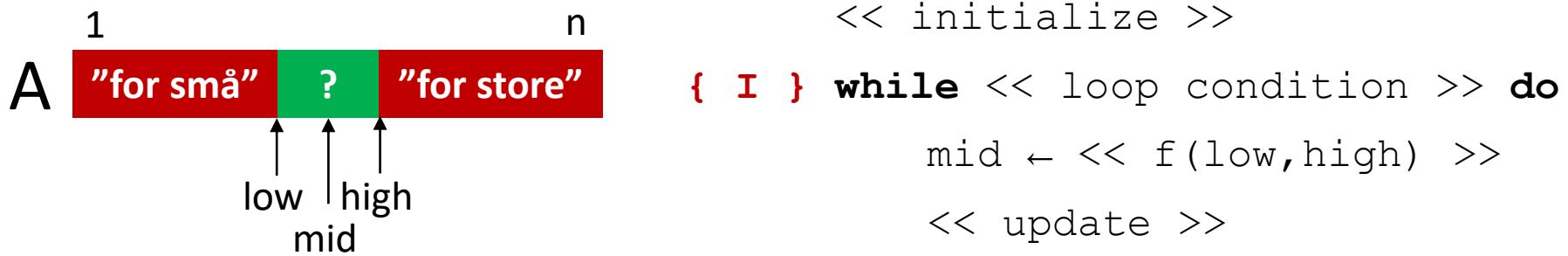
Binær søgning (i sorteret array)



<< initialize >>

{ I } **while** << loop condition >> **do**
 $\text{mid} \leftarrow << f(\text{low}, \text{high}) >>$
 << update >>

Binær søgning (i sorteret array)



I_1 : $(A[i] < x \text{ for } 1 \leq i \leq \text{low}) \wedge (x < A[i] \text{ for } \text{high} \leq i \leq n)$

I_2 : $(A[i] < x \text{ for } 1 \leq i \leq \text{low}) \wedge (x \leq A[i] \text{ for } \text{high} \leq i \leq n)$

I_3 : $(A[i] < x \text{ for } 1 \leq i < \text{low}) \wedge (x \leq A[i] \text{ for } \text{high} < i \leq n)$

I_4 : $(A[i] < x \text{ for } 1 \leq i < \text{low}) \wedge (x \leq A[i] \text{ for } \text{high} \leq i \leq n)$

...

<< initialize >>

- a) $\text{low} \leftarrow 0; \text{high} \leftarrow n+1$
- b) $\text{low} \leftarrow 1; \text{high} \leftarrow n$
- c) $\text{low} \leftarrow 0; \text{high} \leftarrow n$
- d) $\text{low} \leftarrow 1; \text{high} \leftarrow n+1$
- e) ved ikke

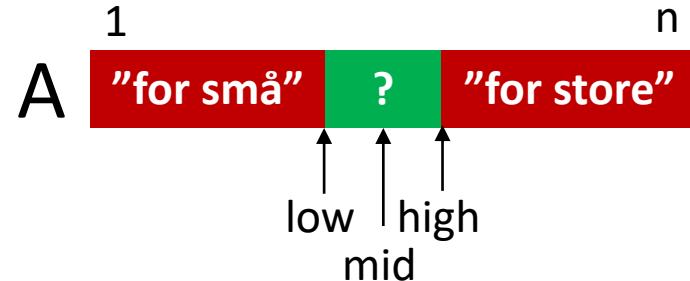
<< loop condition >>

- a) low = high
- b) low ≠ high
- c) low < high
- d) low ≤ high
- e) low > high
- f) ved ikke

<< update >>

- a) if $A[\text{mid}] < x$ then $\text{low} \leftarrow \text{mid}$ else $\text{high} \leftarrow \text{mid}$
- b) if $A[\text{mid}] < x$ then $\text{low} \leftarrow \text{mid}+1$ else $\text{high} \leftarrow \text{mid}$
- c) if $A[\text{mid}] < x$ then $\text{low} \leftarrow \text{mid}$ else $\text{high} \leftarrow \text{mid}+1$
- d) if $A[\text{mid}] < x$ then $\text{low} \leftarrow \text{mid}+1$ else $\text{high} \leftarrow \text{mid}-1$
- e) ved ikke

Binær søgning (i sorteret array)



```
    low ← 1; high ← n
{ I } while low ≤ high do
    mid ← ⌊low + (high - low) / 2⌋
    if A[mid] < x then
        low ← mid + 1
    else
        high ← mid - 1
```

I: $(A[i] < x \text{ for } 1 \leq i < \text{low}) \wedge (x \leq A[i] \text{ for } \text{high} < i \leq n)$

Hvor står svaret ?

a) low - 1



b) low

c) low + 1

d) high - 1

e) high



f) high + 1

g) ved ikke

Algoritme POWER(x, p)

Inputbetingelse : Heltal $x \geq 0$ og $p \geq 0$

Outputkrav : $r = x^p$

Metode : $r \leftarrow 1;$

{ I } **while** $p \geq 1$ **do**

if p lige **then**

$x \leftarrow x * x; p \leftarrow p/2$

else

$r \leftarrow r * x; p \leftarrow p - 1$

p_0 = værdien af p i starten

	Ja	Nej
$p \leq p_0$	<input checked="" type="checkbox"/>	<input type="checkbox"/>
$x^p = r$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$x \leq x_0$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$r \cdot x_0^{p_0} = x^p$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
$x_0^{p_0} = r \cdot x^p$	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Algoritme LOG2(n)

Inputbetingelse : Heltal $n \geq 2$

Outputkrav : $r = \text{intlog}(n) = \lfloor \log_2 n \rfloor$

Metode : $i \leftarrow 1;$

$r \leftarrow 1;$

$p \leftarrow 2;$

{ I } **while** $2p \leq n$ **do**

if $p * p \leq n$ **then**

$p \leftarrow p * p;$

$r \leftarrow 2 * r$

else

$p \leftarrow 2 * p;$

$r \leftarrow r + 1$

	Ja	Nej
$1 \leq r < p$	<input checked="" type="checkbox"/>	<input type="checkbox"/>

$2p \leq n$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
-------------	--------------------------	-------------------------------------

$p = 2^r$	<input checked="" type="checkbox"/>	<input type="checkbox"/>
-----------	-------------------------------------	--------------------------

$p = 2r$	<input type="checkbox"/>	<input checked="" type="checkbox"/>
----------	--------------------------	-------------------------------------

$p = 2^{\text{intlog}(p)}$	<input checked="" type="checkbox"/>	<input type="checkbox"/>
----------------------------	-------------------------------------	--------------------------

Invarianter

- Værktøj til analyse af tilstandene i en løkke i en algoritme
- Designværktøj ”Invariant → kode”
- Kan indfange essentielle egenskaber ved en datastrukturs tilstand (f.eks. heap-order i en binær heap, eller rød-sort træ egenskaber)