Algoritmer og invarianter

Iterative algoritmer

Algoritmen er overordnet set een eller flere **while** eller **for**-løkker.

Iterative algoritmer

Algoritmen er overordnet set een eller flere **while** eller **for**-løkker.

Eksempel: Sekventiel søgning (i usorteret liste):

$i = 1$
while element $i \neq$ mål og $i < |\text{liste}|$:
    $i = i + 1$
if element $i =$ mål:
    then rapporter at mål findes på plads i
    else rapporter at mål ikke findes i liste

Iterative algoritmer

Algoritmen er overordnet set een eller flere **while** eller **for**-løkker.

Eksempel: Sekventiel søgning (i usorteret liste):

$i = 1$
while element $i \neq$ mål og $i < |\text{liste}|$:
    $i = i + 1$
if element $i =$ mål:
    then rapporter at mål findes på plads i
    else rapporter at mål ikke findes i liste

Eksempel: Sekventiel søgning (i stigende sorteret liste):

$i = 1$
while element $i <$ mål og $i < |\text{liste}|$:
    $i = i + 1$
if element $i =$ mål:
    then rapporter at mål findes på plads i
    else rapporter at mål ikke findes i liste

Iterative algoritmer

Eksempel: InsertionSort:

for $i = 1$ til $|\text{liste}| - 1$:
    $j = i + 1$
    while j'te element $<$ $(j-1)$'te element:
        ombyt j'te og $(j-1)$'te element
        $j = j - 1$

Rekursive algoritmer

En algoritmen som kalder sig selv (med et mindre input).

Rekursive algoritmer

En algoritmen som kalder sig selv (med et mindre input).

Eksempel: Binær søgning:

BinSearch(liste)

if |liste| == 0:
then
mål findes ikke i liste
else
m = |liste|/2
if m'te element = mål:
then
mål findes på denne plads.
else
if mål < denne plads:
then BinSearch (liste fra 1 til $m-1$)
else BinSearch(liste fra $m+1$ til |liste|)

Iteration vs. rekursion

Ofte kan en algoritme udtrykkes på begge måder.

Eksempler:

Iteration vs. rekursion

Ofte kan en algoritme udtrykkes på begge måder.

Eksempler:

- Rekursiv sekventiel søgning.

Iteration vs. rekursion

Ofte kan en algoritme udtrykkes på begge måder.

Eksempler:

- Rekursiv sekventiel søgning.
- Rekursiv InsertionSort.

Iteration vs. rekursion

Ofte kan en algoritme udtrykkes på begge måder.

Eksempler:

- Rekursiv sekventiel søgning.
- Rekursiv InsertionSort.
- Iterativ binær søgning.

Analyse af algoritmer

- Korrekthed (giver de altid det rigtige svar?)
- Køretid (hvor lang tid bruger de på at svare?)

Invarianter

Metode til at argumentere om korrekthed for iterative algoritmer. En invariant er et udsagn, som beskriver *state* (dvs. værdien af alle variable) undervejs under algoritmens udførsel.

Invarianter

Metode til at argumentere om korrekthed for iterative algoritmer. En invariant er et udsagn, som beskriver *state* (dvs. værdien af alle variable) undervejs under algoritmens udførsel.

- Man skal kunne argumentere for, at udsagnet altid gælder. Både før første iteration, og efter hver ny iteration.

Invarianter

Metode til at argumentere om korrekthed for iterative algoritmer. En invariant er et udsagn, som beskriver *state* (dvs. værdien af alle variable) undervejs under algoritmens udførsel.

- Man skal kunne argumentere for, at udsagnet altid gælder. Både før første iteration, og efter hver ny iteration.
- Når algoritmen stopper, skal udsagnet gøre det nemt at se, at output er korrekt.

Invarianter

Eksempler:

Invarianter

Eksempler:

- Sekventiel søgning: Målet findes ikke blandt den allerede gennemsøgte del af listen.

Invarianter

Eksempler:

- Sekventiel søgning: Målet findes ikke blandt den allerede gennemsøgte del af listen.
- InsertionSort: Hvis man ser bort fra j'te element, er 1'ste til $(i+1)$'te element sorteret, og det j'te element er $<$ end de $(j+1)$'te til $(i+1)$'te element.

Invarianter

Eksempler:

- Sekventiel søgning: Målet findes ikke blandt den allerede gennemsøgte del af listen.
- InsertionSort: Hvis man ser bort fra j'te element, er 1'ste til $(i+1)$'te element sorteret, og det j'te element er $<$ end de $(j+1)$'te til $(i+1)$'te element.
- Binær søgning (iterativ version): hvis målet findes, er det mellem element i og j.

Invarianter

Eksempler: Euclids algoritme for største fælles divisor af M og N:

$m =$ max af M og N
$n =$ min af M og N
while n ikke går op i m
    $m = n$
    $n =$ rest ved division af n op i m
returner n som svar

Invarianter

Eksempler: Euclids algoritme for største fælles divisor af M og N:

$m =$ max af M og N
$n =$ min af M og N
while n ikke går op i m
    $m = n$
    $n =$ rest ved division af n op i m
returner n som svar

Korrekthed?

Invarianter

Eksempler: Euclids algoritme for største fælles divisor af M og N:

$m =$ max af M og N
$n =$ min af M og N
while n ikke går op i m
    $m = n$
    $n =$ rest ved division af n op i m
returner n som svar

Korrekthed? Ikke oplagt...

Invarianter

Eksempler: Euclids algoritme for største fælles divisor af M og N:

$m =$ max af M og N
$n =$ min af M og N
while n ikke går op i m
    $m = n$
    $n =$ rest ved division af n op i m
returner n som svar

Korrekthed? Ikke oplagt...

Invariant: største fælles divisor af m og n = største fælles divisor af M og N