3.3 Kompleksitet: *tids*-kompleksitet.

Eksempel: kompleksitet af algoritme lineær søgning

Algoritme 2: lineær søgning Side 196

---

procedure $linear\ search$(x:heltal, $a_1, \ldots, a_n$: forskellige heltal)
$i := 1$
while $i \leq n$ and $x \neq a_i$
    $i := i + 1$
if $i \leq n$ **then** $location := i$
else $location := 0$
return $location$
{ hvis $location = 0$ så er x ikke i listen, ellers er $a_{location} = x$ }

$f(n) =$ den tid det tager at finde i liste af n tal ved lineær søgning i værste tilfælde (worst-case): når vi søge gennem hele listen.

average-case

Lettere beregning (uafhængig af hvilken computer der bruges):

$f(n) =$ antal "operationer" der bruges

"operationer" kan eventuelt begrænses til sammenligninger, da det er den vigtigste operation i denne algoritme, og det er den der foretages flest gange, i den konkrete algoritme.

Ikke nødvendigt at finde $f(n)$ eksakt.

Nok at vise at $f(n)$ er $O(g(n))$. (Eller rettere $\Theta(g(n))$.)

Kompleksiteten er så: $O(g(n))$.

procedure $linear\ search(x$:heltal, $a_1, \ldots, a_n$: forskellige heltal)
$i := 1$
while $i \leq n$ 1 and $x \neq a_i$ 2
$i := i + 1$
if $i \leq n$ **then** $location := i$
else $location := 0$
return $location$
{ hvis $location = 0$ så er x ikke i listen, ellers er $a_{location} = x$ }

Antal sammenligninger:

1: højst $n+1$

2: højst n

3 1

I alt: $2n+2$

Operation, der udføres 1 gang.

$2n+2$ er $O(n)$

Kompleksitet: $O(n)$

Addioner

højst n

procedure $binary\ search(x$: heltal, $a_1, \ldots, a_n$: voksende følge af heltal)

$i := 1$

$j := n$

while $i < j$

  $m := \lfloor (i + j)/2 \rfloor$

  if $x > a_m$ **then** $i := m + 1$

  else $j := m$

if $x = a_i$ **then** $location := i$

else $location := 0$

return $location$

$\{$ hvis $location = 0$ så er x ikke i listen, ellers er $a_{location} = x$ $\}$

Hvert gennemløb af While: listen halveres.

Antal gennemløb: $\lceil \log n \rceil$

Hvert gennemløb: konstant antal operationer

Kompleksitet: $O(\log n)$

procedure $bubblesort(a_1, \ldots, a_n$: reelle tal med $n \geq 2)$
for $i := 1$ **to** $n - 1$
    for $j := 1$ **to** $n - i$
        if $a_j > a_{j+1}$ **then** ombyt a_j og a_{j+1}
$\{ a_1, \ldots, a_n$ er nu i voksende rækkefølge$\}$

Antal sammenligninger

$i=1$: $n-1$

$i=2$: $n-2$

$\vdots$

$i=n-1$: 1

I alt: $1+2+3+\ldots+(n-2)+(n-1) = \frac{(n-1)(1+n-1)}{2} = \frac{(n-1)\cdot n}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$

$1+2+\ldots+k = \frac{k(1+k)}{2}$ sett $k = n-1$

$\frac{1}{2}n^2 - \frac{1}{2}n$ er $O(n^2)$

Kompleksitet: $O(n^2)$

4.1

Division med rest

31 divideret med 7

$$31 = 4 \cdot 7 + 3$$

Divisions (Algoritme)

Hvis $a, d \in \mathbb{Z}$, $d > 0$

så findes entydige tal $q, r \in \mathbb{Z}$ så

$$a = q \cdot d + r \quad , \quad 0 \leq r < d$$

Skrives: $q = a \text{ div } d = a/d$

$r = a \text{ mod } d = a \% d$

i C-program

EKS

Hvis $a = 31$, $d = 7$ $\quad$ $\underline{31 = 4 \cdot 7 + 3}$

så er $q = 4$, $r = 3$

$31 \text{ mod } 7 = 3$, $31 \text{ div } 7 = 4$

Hvis $a = -31$, $d = 7$

$-31 = (-4) \cdot 7 - 3$ $\quad$ men $0 \leq -3 < 7$ ikke oppfyllt

$$-31 = (-5)\cdot 7 + (7-3) = (-5)\cdot 7 + 4 \quad , \quad 0 \le 4 < 7$$

$$-31 \bmod 7 = 4$$

Tal der giver rest 4 modulo 7

$\ldots, -10, -3, 4, 11, 18, 25, \ldots$

Rest 5

$\ldots, -9, -2, 5, 12, \ldots$

Rest 6

$\ldots, -8, -1, 6, 13, 20$

Rest 0

$\dots, -14, -7, 0, 7, 14, \dots,$

Rest 1

$-13, -6, 1, 8, 15$

Rest 2

$\dots -12, -5, 2, 9, 16 \dots$

Rest 3

$\dots -11, -4, 3, 10, 17, \dots$

Listene kaldes restklasser

Hvis $a \bmod d = 0$ så går d op i a

Definition

$a, b \in \mathbb{Z}$, $a \neq 0$

Vi siger a går op i b, skrives $a \mid b$

hvis der findes $c \in \mathbb{Z}$ så $b = a \cdot c$

$7 \mid 35$ da $35 = 7 \cdot 5$

$-4 \mid 12$ da $12 = (-4)(-3)$

Sætning

Hvis $a \mid b$ og $a \mid c$ så er $a \mid (b+c)$

Bevis

$a \mid b$ betyder der findes $s \in \mathbb{Z}$ så $b = a \cdot s$

$a \mid c$ — " — $t \in \mathbb{Z}$ så $c = a \cdot t$

$$b + c = a \cdot s + at = a(s+t)$$

Altså $a \mid (b+c)$. $\square$

DEF

$a, b \in \mathbb{Z}$, $m \in \mathbb{Z}^+$

Vi siger at a er kongruent med b modulo m

skrives $a \equiv b \pmod{m}$

hvis $m \mid a-b$

Sætning

$$a \equiv b \pmod{m}$$

$$\Updownarrow$$

$$a \bmod m = b \bmod m$$

$\Updownarrow$

a og b er i samme restklasse.

Eks

$$9 \equiv 44 \pmod 7$$

Da $9 - 44 = -35 = 7 \cdot (-5)$

$9 = 1 \cdot 7 + 2$ $\quad$ $9 \bmod 7 = 2$

$44 = 6 \cdot 7 + 2$ $\quad$ $44 \bmod 7 = 2$

EKS $44 \equiv -12 \pmod 7$

Da $44-(-12) = 56 = 7 \cdot 8$

$-12 = (-2) \cdot 7 + 2$ $\qquad$ $-12 \bmod 7 = 2$

Setning

Hvis $a \equiv b \pmod{m}$ og $c \equiv d \pmod{m}$

så er

$$a + c \equiv b + d \pmod{m}$$

og $$a \cdot c \equiv b \cdot d \pmod{m}$$

EKS

Udregn $(19^2 + 11 \cdot 15) \bmod 8$

$19 = 2 \cdot 8 + 3$, $\quad 19 \bmod 8 = 3$

$19 \equiv 3 \pmod{8}$

$11 = 1 \cdot 8 + 3$ $\quad 11 \bmod 8 = 3$

$11 \equiv 3 \pmod{8}$

$15 = 2 \cdot 8 - 1$ $\quad 15 \equiv -1 \pmod{8}$

$19^2 + 11 \cdot 15 \equiv 3 \cdot 3 + 3 \cdot (-1) \pmod{8}$

$$= 6$$

$$19^2 + 11 \cdot 15 \equiv 6 \pmod{8}$$

$$\left(19^2 + 11 \cdot 15\right) \bmod 8 = 6$$