Afsnit 1.6

Typer af resultater, der kan bevises:

Sætning (Teorem): vigtigt resultat.

Proposition: mindre vigtigt resultat.

Lemma: resultat, der i sig selv ikke er så interessant, men kan bruges til at bevise en sætning.

Korollar: resultat, der let kan bevises, når en sætning er bevist.

Afsnit 3.1

Definition 1. En algoritme er en endelig mængde af præcise instruktioner til at udføre en beregning eller løse et problem.

Dette er ikke en egentlig definition i matematisk forstand.

Rækkefølgen af instruktioner er vigtig!

Yderligere egenskaber for en algoritme:
input, output, præcis defineret, korrekt, endelig, hver skridt kan udføres på endelig tid, generel

Algoritme=Procedure

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** $location := 0$

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

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

$i := 1$

$j := n$

while $i < j$

begin

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

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

  else $j := m$

end

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

else $location := 0$

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

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$\}$

Sætning (Standseproblemet).
Der findes ikke en procedure $H(P, I)$, der læser en procedure P og et input I til P, og som giver output:

“standser” hvis $P(I)$ standser,

“uendelig løkke” hvis $P(I)$ ikke standser.

Bevis (ved modstrid).
Antag H eksisterer.

Konstruer en ny procedure:

Procedure $K(P)$
if $H(P,P)$ svarer "standser" **then**
    gå i uendelig løkke
else stands

Standser $K(K)$???

Hvis $K(K)$ standser så giver $H(K,K)$ output "standser". Dermed går $K(K)$ i en uendelig løkke.

Hvis $K(K)$ ikke standser så giver $H(K,K)$ output "uendelig løkke". Dermed standser $K(K)$. $\square$

3.3: Store O.

$f(x), g(x)$ funktioner der er defineret når x er et tilstrækkeligt stort helt eller reelt tal. $f(x), g(x) \in \mathbb{R}$.

$f(x)$ er $O(g(x))$ hvis der konstanter C, k så

$$|f(x)| \leq C|g(x)|, \text{ for alle } \ x > k.$$

$f(x)$ er $\Omega(g(x))$ hvis der konstanter $C > 0, k$ så

$$|f(x)| \geq C|g(x)|, \text{ for alle } \ x > k.$$

$$f(x) \text{ er } \Omega(g(x)) \Leftrightarrow g(x) \text{ er } O(f(x)).$$

$f(x)$ er $\Theta(g(x))$ hvis $f(x)$ er $O(g(x))$ og $f(x)$ er $\Omega(g(x))$.

3.3: Kompleksitet af algoritme.

n : mål for størrelsen af input.

$f(n)$: den tid det <u>højst</u> tager for algoritmen at løse problemet med denne inputstørrelse.

$f(n)$ angives normalt ikke eksakt, men med hjælp af store O notation.