

# MAT1030 – Diskret matematikk

## Plenumsregning 1: Kapittel 1

Roger Antonsen

Matematisk Institutt, Universitetet i Oslo

17. januar 2008



# Velkommen til plenumsregning for MAT1030

- Torsdager 10:15–12:00
- Gjennomgang av ukeoppgaver
- Gjennomgang av eksempler fra boka
- Litt repetisjon fra forelesningen
- Spørsmål og svar
- **Forsøk på oppgavene selv først!**

# Repetisjon: Algoritmer og pseudokode

- En **algoritme** er en oppskrift som forteller oss hvordan vi skritt for skritt skal kunne oppnå et resultat eller løse et problem.
- Det skal ikke kreves intelligens eller forståelse for å følge den.
- En **pseudokode** er en måte å beskrive en algoritme på.
- Hvert steg i algoritmen skal være beskrevet på en **entydig** måte.

# Eksempel fra boka

## Eksempel 1.1.1

Skriv en algoritme som regner ut arealet av en sirkel, gitt radiusen.

## Løsning

1. Input  $r$  [ $r$  er radiusen til sirkelen.]
2.  $areal \leftarrow \pi r^2$
3. Output  $areal$

- Hvert steg i algoritmen er nummerert.
- Kommentarer skrives mellom [].
- Symbolet  $\leftarrow$  betegner *tilordning*.

# Repetisjon: Kontrollstrukturer

- En **kontrollstruktur** brukes for å styre hvordan, og hvorvidt, de enkle instruksjonene i en pseudokode skal utføres.

## For-do

1.  $sum \leftarrow 0$
2. **For**  $i = 1$  **to**  $6$  **do**
  - 2.1.  $sum \leftarrow sum + i$

## Kjøring

Steg	$sum$	$i$
-	21	-

# Repetisjon: Kontrollstrukturer

## If-then

1. **If**  $x < 0$  **then**
  - 1.1.  $x \leftarrow -x$

## If-then-else

1. **If**  $x \geq 0$  **then**
  - 1.1.  $y \leftarrow \sqrt{x}$**else**
  - 1.2. Output ' $\sqrt{x}$  fins ikke'

## While-do

1. **While**  $svar \neq 1$  **do**
  - 1.1. Input  $svar$

## Repeat-until

1.  $i \leftarrow 0$
2. **Repeat**
  - 2.1.  $i \leftarrow i + 1$
  - 2.2. Input  $x_i$**until**  $x_i = 0$

# Eksempler fra boka

## Eksempel 1.2.1/1.2.2

Finn det minste tallet i en liste av tall.

# Eksempler fra boka

## Løsning

1. Input the number of values  $n$
2. Input the list of numbers  $x_1, \dots, x_n$
3.  $min \leftarrow x_1$
4. **For**  $i = 2$  **to**  $n$  **do**
  - 4.1. **If**  $x_i < min$  **then**
    - 4.1.1.  $min \leftarrow x_i$
5. Output  $min$



# Eksempler fra boka

## Løsning

1. Input the number of values  $n$
2. Input the list of numbers  $x_1, \dots, x_n$
3.  $min \leftarrow x_1$
4. **For**  $i = 2$  **to**  $n$  **do**
  - 4.1. **If**  $x_i < min$  **then**
    - 4.1.1.  $min \leftarrow x_i$
5. Output  $min$

## Kjøring

Steg	$min$	$i$	$n$	$x_1$	$x_2$	$x_3$
5	4	-	3	5	4	8

# Reserverte ord og uttrykk i pseudokode

- **If**
- **then**
- **else**
- **For**
- **to**
- **do**
- **While**
- **Repeat**
- **until**
- **and**
- **or**
- **not**
- Input
- Output
- $\leftarrow$
- true
- false
- [] (eller {} i boka)
- vanlig matematiske uttrykk (f.eks.  $x < 0$  og  $\sqrt{x}$ )

## Eksempel 1.2.3

Lag en algoritme som sjekker om en streng  $c_1c_2 \dots c_n$  av  $n$  tegn består av kun sifre, eller om ikke-sifre forekommer, og skriv ut en passende melding.

- Hvis input er 12345 skal algoritmen returnere meldingen:  
Strengen inneholder kun sifre.
- Hvis input er 123@45 skal algoritmen returnere meldingen:  
Strengen inneholder tegn som ikke er sifre.

## Utkast 1

1.  $i \leftarrow 1$ ; *ikkesiffer\_oppdaget*  $\leftarrow$  false
  2. **Repeat**
    - 2.1. **If**  $c_i$  ikke er et siffer **then**
      - 2.1.1. *ikkesiffer\_oppdaget*  $\leftarrow$  true
    - 2.2.  $i \leftarrow i + 1$
- until** *ikkesiffer\_oppdaget* = true

- Identifikatorer/variable skrives i *kursiv* uten mellomrom.
- *ikkesiffer\_oppdaget* kalles gjerne for en *logisk* (eller Boolsk) variabel, siden den kun vil ta verdiene true eller false.
- Kunne ha skrevet '**until** *ikkesiffer\_oppdaget*' i siste linje.
- Denne vil ikke **terminere** hvis strengen består av kun sifre.

## Utkast 2

1.  $i \leftarrow 1$ ; *ikkesiffer\_oppdaget*  $\leftarrow$  false
  2. **Repeat**
    - 2.1. **If**  $c_i$  ikke er et siffer **then**
      - 2.1.1. *ikkesiffer\_oppdaget*  $\leftarrow$  true
    - 2.2.  $i \leftarrow i + 1$
- until** *ikkesiffer\_oppdaget* = true **or**  $i = n + 1$

- Hvis alle tegn er sifre vil algoritmen terminere når verdien til  $i$  er  $n + 1$ .
- Nå gjenstår Input og Output.

# Eksempler fra boka

## Utkast 3

1. Input  $n$
2. Input  $c_1 c_2 \dots c_n$
3.  $i \leftarrow 1$ ; *ikkesiffer\_oppdaget*  $\leftarrow$  false
4. **Repeat**
  - 4.1. **If**  $c_i$  ikke er et siffer **then**
    - 4.1.1. *ikkesiffer\_oppdaget*  $\leftarrow$  true
  - 4.2.  $i \leftarrow i + 1$**until** *ikkesiffer\_oppdaget* = true **or**  $i = n + 1$
5. **If** *ikkesiffer\_oppdaget* = true **then**
  - 5.1. Output 'Strengen inneholder tegn som ikke er sifre.'**else**
  - 5.2. Output 'Strengen inneholder kun sifre.'

- Men, hva hvis  $n = 0$ ? Da vil steg 4.1 ikke kunne utføres.

# Kommentarer

- **Repeat-until**-løkken blir alltid utført minst én gang.
- Et alternativ er å bruke en **While**-løkke.
- Da utføres testen i begynnelsen i stedet for på slutten, som med **Repeat-until**.

# Eksempler fra boka

## Løsning

1. Input  $n$
2. Input  $c_1 c_2 \dots c_n$
3.  $i \leftarrow 0$ ;  $ikkesiffer\_oppdaget \leftarrow \text{false}$
4. **While**  $ikkesiffer\_oppdaget = \text{false}$  **and**  $i < n$  **do**
  - 4.1.  $i \leftarrow i + 1$
  - 4.2. **If**  $c_i$  ikke er et siffer **then**
    - 4.2.1.  $ikkesiffer\_oppdaget \leftarrow \text{true}$
5. **If**  $ikkesiffer\_oppdaget = \text{true}$  **then**
  - 5.1. Output 'Strengen inneholder tegn som ikke er sifre.'**else**
  - 5.2. Output 'Strengen inneholder kun sifre.'



# Eksempler fra boka

## Eksempel 1.3.1

Lag en algoritme som regner ut  $x^n$ , hvor  $x$  er et reelt tall og  $n$  er et positivt heltall. (Vi antar at vi har multiplikasjon, men ikke eksponensiering.)

### Løsning

1. Input  $x, n$
2.  $svar \leftarrow x$
3. **For**  $i = 1$  **to**  $n - 1$  **do**
  - 3.1.  $svar \leftarrow svar \times x$
4. Output  $svar$

### Kjøring

$x$	$n$	$i$	$svar$
2	3	-	8

# Eksempler fra boka

## Eksempel 1.3.2

Lag en algoritme som bytter verdien til to variable.

### Feil

1.  $x \leftarrow y$
2.  $y \leftarrow x$

### Riktig

1.  $temp \leftarrow x$
2.  $x \leftarrow y$
3.  $y \leftarrow temp$

### Kjøring

$$\begin{array}{cc} x & y \\ \hline 2 & 3 \end{array} \rightsquigarrow \begin{array}{cc} x & y \\ \hline 3 & 3 \end{array}$$

### Kjøring

$$\begin{array}{ccc} x & y & temp \\ \hline 3 & 2 & 2 \end{array}$$

# Eksempler fra boka

## Eksempel 1.3.3

Søkere til en jobb tar en test med 20 spørsmål. Lag en algoritme som returnerer en liste av jobbsøkere (identifisert med tall), deres poengsum og en beskjed om hvorvidt de vurderes for stillingen (de som har mer enn 16 poeng) eller settes på venteliste (de med poengsum fra 12 til 15).

- Vi bruker *antall\_søkere* for å betegne antall søkere.
- Vi bruker en **For-do**-løkke, fra 1 til *antall\_søkere*, og gjør det samme for hver søker.
- De 20 svarene som søker *i* har avgitt betegner vi med  $a_{i,1}, a_{i,2}, \dots, a_{i,20}$ .
- Det riktige svaret på spørsmål *q* betegner vi med  $c_q$ .
- Vi bruker enda en **For-do**-løkke for å sjekke svarene.

## Løsning

1. Input *antall\_søkere*
2. **For**  $i = 1$  **to** *antall\_søkere* **do**
  - 2.1.  $score \leftarrow 0$
  - 2.2. **For**  $q = 1$  **to** 20 **do**
    - 2.2.1. Input  $a_{i,q}$  [ $a_{i,q}$  er søker  $i$ s svar på spørsmål  $q$ .]
    - 2.2.2. **If**  $a_{i,q} = c_q$  **then**
      - 2.2.2.1.  $score \leftarrow score + 1$
  - 2.3. Output  $i, score$
  - 2.4. **If**  $score \geq 16$  **then**
    - 2.4.1. Output 'Anbefalt'
    - else if**  $score \geq 12$  **then**
    - 2.4.2. Output 'Venteliste'

# En praktisk forkortelse

## Pseudokode

1. **If**  $x > 0$  **then**
  - 1.1. Output 'Større'
- else**
  - 1.2. **If**  $x < 0$  **then**
    - 1.2.1. Output 'Mindre'

## Forkortet pseudokode

1. **If**  $x > 0$  **then**
  - 1.1. Output 'Større'
- else if**  $x < 0$  **then**
  - 1.2. Output 'Mindre'