

# Forelesning 1

## Algoritmer, pseudokoder og kontrollstrukturer

Dag Normann - 14. januar 2008

### Vi som skal undervise

- Dag Normann
  - Roger Antonsen
  - Christian Schaal
  - Robin Bjørnetun Jacobsen
- <http://www.uio.no/studier/emner/matnat/math/MAT1030/v08/>

### Hva er diskret matematikk?

- Diskret matematikk er et samlebegrep for matematikk hvor kontinuitet, geometri eller algebra ikke spiller noen stor rolle.
- Diskret matematikk er matematikken tilpasset en digital verden, mens mye annen matematikk er tilpasset en analog verden.
- I MAT1030 skal vi ta for oss temaer som er relevante for en grunnleggende forståelse av bruk av, og virkemåte til, datamaskiner.

### Hva er innholdet i MAT1030?

- Algoritmer: Hva og hvordan.
- Representasjon av tall.
- Hvordan representeres tall og i en datamaskin.
- Logikk:
  - Relevans for informatikk.
  - Innføring i utsagnslogikk (Boolesk logikk).
  - Litt om bevisteknikker.
- Mengdelære:
  - Grunnleggende begreper
  - Relasjoner.
  - Funksjoner.
- Induksjon og rekursjon.
- Kombinatorikk.
- Grafteori med anvendelser.
- Litt om trær.
- Kompleksitet av algoritmer.

## Hva legger vi vekt på?

- Hvilken relevans har dette stoffet for studier i informatikk?
- Hvordan kan vi finne algoritmer for å løse de problemene som presenterer seg?

Vi kommer til å arbeide med algoritmer i tilknytning til logikk i større grad enn det boka gjør, og vi vil utvikle en del ekstra undervisningsmateriale underveis.

## Før vi begynner

- MAT1030 er under omlegging, og vi trenger en kontinuerlig tlbakemelding på årets opplegg.
- Vi trenger to eller flere kontaktpersoner vi kan ha møter med gjennom semesteret.
- Vi skal gjennomføre en emneevaluering i samarbeid med kontaktpersonene.
- Valg av kontaktpersoner finner sted neste onsdag.

## Noen viktige datoer

- 29.02.2008: Innlevering av Oblig 1.
- 25.04.2008: Innlevering av Oblig 2.
- 10.06.2008, 09.00 - 12.00: Eksamen.

# Viktig informasjon til alle studenter

## Husk:

**1. februar** er siste frist for:

- betaling av semesteravgift
- semesterregistrering
- melding til eksamen
- å søke om tilrettelegging ved eksamen

Dersom du har gjort alt riktig skal det se slik ut i studentweb:

<b>Betalt semesteravgift:</b>	Vår 2008
<b>Semesterregistrert:</b>	Vår 2008

Dersom det ikke står "Vår 2008" under henholdsvis "Betalt semesteravgift" og "Semesterregistrert" risikerer du å ikke få ta eksamen og å miste din studierett.

Har du problemer, ta kontakt med MN studieinfosenter i Vilhelm Bjerknes' hus FØR 1. februar.  
Telefon: 22 85 63 44  
e-post: [studieinfo@matnat.uio.no](mailto:studieinfo@matnat.uio.no)

## ALGORITMER

En algoritme er en oppskrift som forteller oss hvordan vi skritt for skritt skal kunne oppnå et resultat eller løse et problem.

Eksempler på algoritmer kan være:

- Kakeoppskrifter.
- Automatisk innsjekking på fly.
- Beskrivelsen av hvordan man utfører divisjon mellom flersifrede tall.
- Oppskrift på hvordan man løser opp parenteser og trekker sammen flerleddede uttrykk i algebra.

## Algoritmer

Hva er det som kjennetegner en algoritme?

Det skal ikke kreves intelligens eller forståelse for å følge den.

- Du skal ikke kunne kjemi for å bake en kake.
- Du skal kunne sjekke inn på fly selv om du har teknologifobi.
- Det er ikke nødvendig å forstå hva man gjør når man utfører en divisjon, regnetrening er det som trengs.
- Mange lærer seg hvordan de kan løse oppgaver i skolealgebra, uten å ha peiling på hva de egentlig driver med.

## Algoritmer

Vi skal fokusere på algoritmer som

- beregner funksjoner
- avgjør om et objekt eller en datamengde har en gitt egenskap eller ikke
- organiserer gitte data på en ønsket måte (eksempelvis ordner dataene)
- utfører andre oppgaver i tilknytning til matematikk eller informatikk som vi ønsker å kunne få utført.

## Algoritmer

Hvem ønsker vi å kommunisere algoritmen til, og hvordan skal det gjøres?

1. Kakebakere, flypassasjerer og liknende.
2. Skolebarn/ungdom og studenter som skal lære matematikk.
3. Teknisk kyndige medmennesker som skal "forstå" algoritmen.
4. Datamaskiner som skal utføre algoritmen for oss.

I MAT1030 er gruppe 3 den mest aktuelle.

## PSEUDOKODER

En pseudokode er en måte å beskrive en algoritme på.

Pseudokoden beskriver hvordan algoritmen kan følges trinn for trinn.

En pseudokode formuleres i et språk som er mer teknisk enn naturlige språk og mindre teknisk enn programmeringsspråk.

Vi skal bruke pseudokoder på samme måte som i læreboka.

Man må se eksempler, og øve, for å bli flink til å skrive pseudokoder.

## Pseudokoder

### Eksempel (Areal av trekant)

1. Input  $h$  [ $h$  er høyden i trekanten.]
2. Input  $g$  [ $g$  er lengden på grunnlinjen i trekanten.]
3.  $areal \leftarrow \frac{h \cdot g}{2}$
4. Output  $areal$

## Pseudokoder

Vi kunne ha skrevet en annen pseudokode for å beregne det samme:

### Eksempel

1. Input  $h$
2. Input  $g$
3.  $areal \leftarrow h \cdot g$
4.  $areal \leftarrow \frac{areal}{2}$
5. Output  $areal$

## Pseudokoder

Så langt består en pseudokode av en nummerert liste instruksjoner hvor hver instruksjon har et av følgende tre formater:

- Gi en input-verdi til en variabel.
- Gi en variabel en ny verdi, som en funksjon av de eksisterende verdiene på variablene.
- Gi verdien til en av variablene som output, det vil si resultatet av algoritmen.

Vi kan bruke hva vi vil som variable, eksempelvis er  $h$ ,  $g$  og  $areal$  variablene i pseudokodene vi har sett på.

## Pseudokoder

Hvis vi skal beregne verdien av en formel for areal, volum, hastighet etter en viss tids fritt fall og liknende, kan vi bruke pseudokoder slik vi har sett dem til nå.

Det finnes imidlertid algoritmer, og tilhørende kontrollstrukturer, for å beregne

- $|x|$  fra  $x$
- $n!$  fra  $n$
- Ledd nummer  $n$  i Fibonacci-følgen

1, 1, 2, 3, 5, 8, 13, 21, . . . .

og for å undersøke om

- Parentesene i et algebraisk uttrykk er satt på lovlig måte
- Et naturlig tall er et primtall

## KONTROLLSTRUKTURER

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

Vi skal innføre de kontrollstrukturene det er aktuelt å bruke i dette emnet via eksempler på bruk. Disse eksemplene skal supplere eksemplene fra læreboka.

Vi skal benytte oss av de samme kontrollstrukturene som boka.

### Kontrollstrukturer

#### Eksempel (Absoluttverdi)

Vi skal gi en pseudokode for å beregne absoluttverdien til et tall  $x$ :

```
1 Input  $x$ 
2 If  $x < 0$  then
    2.1  $x \leftarrow -x$ 
3 Output  $x$ 
```

### Kontrollstrukturer

#### Eksempel (Avstand)

Vi skal gi en pseudokode for å beregne avstanden mellom to heltall.

```
1 Input  $n$  [ $n$  et heltall]
2 Input  $m$  [ $m$  et heltall]
3 If  $n < m$  then
    3.1  $x \leftarrow m - n$ 
else
    3.2  $x \leftarrow n - m$ 
4 Output  $x$ 
```

## Kontrollstrukturer

Vi skal se tre eksempler på hvordan vi kan skrive pseudokoder for algoritmer som beregner

$$n! = 1 \cdot 2 \cdot \dots \cdot n.$$

I det ene eksemplet bruker vi en while-løkke, i det andre en repeat-until-løkke og i det siste en for-løkke.

## Kontrollstrukturer

### Eksempel (while-løkke)

```
1 Input  $n$  [ $n \geq 1$ ,  $n$  heltall]
2  $x \leftarrow 1$ 
3  $i \leftarrow 1$ 
4 While  $i \leq n$  do
    4.1  $x \leftarrow x \cdot i$ 
    4.2  $i \leftarrow i + 1$ 
5 Output  $x$ 
```

## Kontrollstrukturer

### Eksempel (repeat-until-løkke)

```
1 Input  $n$  [ $n \geq 1$ ,  $n$  heltall]
2  $x \leftarrow 1$ 
3  $i \leftarrow 1$ 
4 Repeat
    4.1  $x \leftarrow x \cdot i$ 
    4.2  $i \leftarrow i + 1$ 
until  $i = n + 1$ 
5 Output  $x$ 
```

## Kontrollstrukturer

### Eksempel (for-løkke)

```
1 Input  $n$  [ $n \geq 1$ ,  $n$  heltall]
2  $x \leftarrow 1$ 
3  $i \leftarrow 1$ 
4 For  $i = 1$  to  $n$  do
    4.1  $x \leftarrow x \cdot i$ 
5 Output  $x$ 
```

## Kontrollstrukturer

Bytte av verdi på variablene, hvordan vi ikke skal gjøre det og hvordan vi skal gjøre det.

### Eksempel (Feil måte)

```
1  $y \leftarrow x$   
2  $x \leftarrow y$ 
```

### Eksempel (Riktig måte)

```
1  $hjelp \leftarrow x$   
2  $x \leftarrow y$   
3  $y \leftarrow hjelp$ 
```