



**Høgskolen i Gjøvik**  
Avdeling for teknologi

---

# E K S A M E N

**FAGNAVN:** Algoritmiske metoder (I)

**FAGKODE:** L189A og L167E

**EKSAMENS DATO:** 9. desember 2002

**KLASSE(R):** 01HIND\* / 01HINE\* / div. andre

**TID:** L189A: 09.00-14.00  
L167E: 09.00-13.00

**FAGLÆRER:** Frode Haug

**ANTALL SIDER UTLEVERT:** 4 (inkludert denne forside)

**TILLATTE HJELPEMIDLER:** Alle trykte og skrevne.

- Kontroller at alle oppgavearkene er tilstede.
- Innføring med penn, eventuelt trykkblyant som gir gjennomslag. Pass på så du ikke skriver på mer enn ett innføringsark om gangen (det blir uleselige gjennomslag når flere ark ligger oppå hverandre).
- Ved innlevering skilles hvit og gul besvarelse, som legges i hvert sitt omslag.
- Oppgavetekst, kladd og blå kopi beholder kandidaten til klagefristen er over.
- Ikke skriv noe av din besvarelse på oppgavearkene. Men, i oppgavetekst der du skal fylle ut svar i tegning/tabell/kurve, skal selvsagt dette innleveres sammen med hvit besvarelse.
- Husk kandidatnummer på alle ark. Ta vare på dette nummeret til sensuren faller.

**NB: De som tar L189A (3 vekttall) skal *løse alle* oppgavene.**

**Oppgavene er da vektlagt med: 25%, 25%, 30% og 20%**

**De som tar L167E (2 vekttall) skal *ikke løse* oppgave nr. 2.**

**Oppgavene er da vektlagt med: 40%, 35% og 25%**

## Oppgave 1 (teori)

Denne oppgaven inneholder tre uavhengige oppgaver fra kap. 8, 12 og 15 i læreboka.

- a) Vi skal utføre Shellsort på key'ene "STROGANOFF". Jfr. kode s.109 i læreboka.  
For hver gang indre for-løkke er ferdig (etter:  $a[j] = v$ ):  
**Tegn opp arrayen og skriv verdiene til 'h' og 'i' underveis i sorteringen.**  
**Marker spesielt de key'ene som har vært involvert i sorteringen** (jfr. fig.8.7 s.108).
  
- b) Vi skal utføre rekursiv Mergesort på key'ene "STROGANOFF". Jfr. kode s.166 i læreboka.  
For hver gang tredje og siste for-løkke i koden s.166 er ferdig:  
**Tegn opp arrayen med de key'ene som har vært involvert i sorteringen** (jfr. fig.12.1 s.167).
  
- c) Legg key'ene "BIFFSTROGANOFF" (i denne rekkefølge fra venstre til høyre) inn i et 2-3-4 tre. **Tegn opp treet etterhvert som bokstavene legges inn.**  
**Gjør også om sluttresultatet til et Red-Black tre.**

## Oppgave 2 (teori) - Skal *ikke* løses av de som tar L167E (2 vekttall)

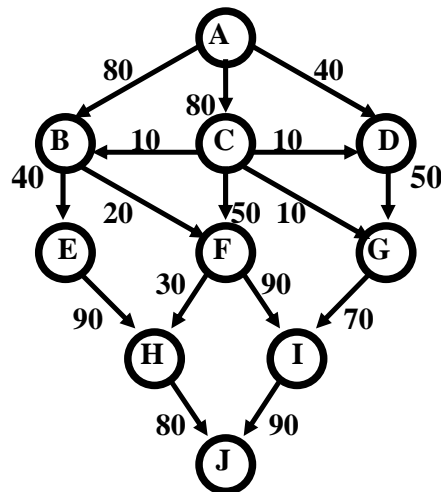
Denne oppgaven inneholder tre uavhengige oppgaver fra kap. 30 & 33 i lærebok og om FSM.

- a) Følgende kanter i en (ikke-rettet, ikke-vektet) graf er gitt:

AD GB CF DB AE BE GD BF FD CA

Vi skal nå utføre union-find på denne grafen. Tegn opp arrayen “dad”s innhold etterhvert som skogen bygges opp (jfr. fig.30.6) vha. “find”-funksjonen s.444 i læreboka. Ut fra dette: tegn også opp den resulterende union-find skogen (jfr. nedre høyre skog i fig.30.5).

- b) Vi har følgende nettverk (dvs. en graf som er både vektet og rettet):



'A' er kilden og 'J' er kummen. Kapasiteten for hvert enkelt rør/kant er skrevet på figuren, og flyten i røret går i pilretningen. Tegn figuren opp igjen, men nå med ny verdi på hver kant, som forteller hva som er max. flyt langs hver av kantene (fra kilde til kum).

- c) Følgende deterministiske endelige tilstandsmaskin (FSM) er gitt:

$$M = ( \{ q_0, q_1, q_2, q_3 \}, \{ a, b \}, \delta, q_0, \{ q_2 \} )$$

|                      |                        |                        |
|----------------------|------------------------|------------------------|
| Transisjonstabellen: | $\delta(q_0, a) = q_1$ | $\delta(q_0, b) = q_0$ |
|                      | $\delta(q_1, a) = q_3$ | $\delta(q_1, b) = q_2$ |
|                      | $\delta(q_2, a) = q_2$ | $\delta(q_2, b) = q_2$ |
|                      | $\delta(q_3, a) = q_3$ | $\delta(q_3, b) = q_0$ |

Tegn tilstandsmaskinen.

Hva slags setninger/språk godtar den ?

## Oppgave 3 (koding)

Denne oppgaven omhandler binære trær, der hver node i treet er definert på følgende måte:

```
struct node {  
    char key;           // Nodens ID/navn.  
    node *l, *r;        // Pekere til venstre og høyre subtre/barn.  
};
```

Du skal **lage tre ulike rekursive funksjoner** som finner ut noe om egenskapene til et slikt binært tre:

- a) **Lag funksjonen** `bool bst(node * t);`  
Funksjonen skal returnere true/false til om (sub)treet tilpekt av 't' er et **binært søketre eller ei**.
- b) **Lag funksjonen** `..... ft(node * t, .....);`  
Funksjonen skal finne ut om (sub)treet tilpekt av 't' er et **fullt tre eller ei**.  
(Husk at treet er 'fullt' når *alle* blad-nodene slutter på *eksakt* samme nivå. Dvs. at *alle* nodene har like mange barn på venstre som på høyre side.)
- c) **Lag funksjonen** `..... kt(node * t, .....);`  
Funksjonen skal finne ut om (sub)treet tilpekt av 't' er et **komplett tre eller ei**.  
(Husk at treet er 'komplett' når *alle* blad-nodene (og z-nodene) slutter på *samme nivå +/- 1*, og at det er fylt opp fra venstre mot høyre på det siste nivået.)

I *alle de tre funksjonene* skal du legge vekt på at de er effektive, ved at de terminerer rekursjonen når de oppdager at treet ikke kan være av aktuell type.

**Ad oppgave b) og c):** Finn selv ut hva det er hensiktsmessig at funksjonene returnerer, hva de bør ha av andre parametre og/eller hvilke globale variable de skal operere på.

## Oppgave 4 (koding)

En fil ("KODE.CPP") inneholder masse ferdigskrevet C++ -kode, men programmereren har *ikke* laget innrykk på *noen* av linjene. Dvs. all koden på hver linje er skrevet helt inntil venstre marg/kant og mot høyre. *Alle* krøllparanteser ( '{ ' og ' } ' ), som omslutter blokker, *står alltid aller først på linja*.

(Alt dette kan du altså få forutsette at er korrekt på filen.)

Du skal i denne oppgaven **lage et komplett program som leser innholdet fra/på denne filen, og som lager en pen utskrift av det** til filen ("KODE2.CPP"). Med "pen" menes at alle linjer inneholdende matchende krøllparanteser skal stå i samme kolonne som linjen som de utgjør blokken for, og alle linjer inne i vedkommende blokk skal være innrykket to blanke mot høyre ift. krøllparantesene. **Løsningen din skal bl.a. basere seg på en rekursiv funksjon som utfører det meste av dette arbeidet.**

Lykke til !

frode@haug.com