

## EKSAMENSOPPGAVE

**Fag:** IAD20003 – Algoritmer og datastrukturer  
**Lærer:** André Hauge

|                                                                       |                                |                       |
|-----------------------------------------------------------------------|--------------------------------|-----------------------|
| <b>Grupper: D2A</b>                                                   | <b>Dato: 21.12.2004</b>        | <b>Tid: 0900-1300</b> |
| <b>Antall oppgavesider: 5 med forside</b>                             | <b>Antall vedleggssider: 0</b> |                       |
| <b>Hjelpemidler: Alle trykte og skrevne hjelpemidler, kalkulator.</b> |                                |                       |
| <b>Sensurfrist: 17.01.2005</b>                                        |                                |                       |
| <b>KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG</b> |                                |                       |

### Råd for oppgaveløsningen:

- Les alle oppgavene før du starter.
- Dersom du finner at oppgaveteksten ikke gir tilstrekkelig informasjon, eller at oppgaveteksten er tvetydig, så må du gjøre dine egne forutsetninger. Disse må du i så fall presisere og begrunne.
- Dersom du får knapp tid, er det bedre å lage skisser på punktene du ikke får tid til fremfor å la punktene være blanke.
- Dersom du har mye tid igjen, ikke gå før tiden, disse 4 timene og hvordan du benytter deg av dem bestemmer din karakter i faget.
- Husk at begrunnelsen og beskrivelsen i svarene er svært viktig og derfor bør være klar.

Karaktersetting av eksaminandens besvarelse foretas etter en helhetsvurdering.

## Del 1 – Totalt 20%

Denne oppgaven omhandler forskjellige typer binære trær.

### Oppgave 1.1 - 5%

Bygg et binært søketre av de tallene gitt under. Tallene settes inn i treet i den rekkefølgen de er gitt. Tallene er separert med mellomrom.

40 12 5 35 1 87 16 52 120

Tegn treet og forklar hvorfor det ble slik det ble.

### Oppgave 1.2 - 5%

En heap er et binært tre som har visse spesielle karakteristikk. Beskriv hva en heap er.

### Oppgave 1.3 - 5%

Tegn en heap (det binære treet) basert på dataene som ligger i figur 1.3.

|   |   |   |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|----|----|----|---|----|----|----|----|----|----|----|----|----|----|----|
| 3 | 7 | 5 | 17 | 12 | 20 | 9 | 18 | 23 | 16 | 20 | 27 | 60 | 71 |    |    |    |    |
| 0 | 1 | 2 | 3  | 4  | 5  | 6 | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |

Figur 1.3 – Tabell implementasjon av heap

### Oppgave 1.4 - 5%

Dersom vi foretar en **deleteMin** operasjon på heapen du tegnet i oppgave 3.3. Hvordan vil treet se ut etter at operasjonen er fullført? Tegn og forklar hva som skjer.

## Slutt del 1

## **Del 2 – Totalt 20%**

Nedenfor er det gitt tre små programdeler skrevet i Java-kode. Disse heter alg1, alg2 og alg3. Hva disse metodene gjør er uviktig, i oppgaven som følger er det kjøretiden som skal vurderes.

### **Alg1**

```
void alg1(int tall) {  
    if( tall >= 2 )  
        alg1( tall/2 );  
    System.out.print(tall%2);  
}
```

### **Alg2**

```
void alg2(int[] arr) {  
    int i = 0;  
  
    while(i < arr.length) {  
        i++;  
    }  
    for(int j = i; j > 0; j--) {  
        System.out.println(arr[j]);  
    }  
}
```

### **Alg 3**

```
void alg3(int n) {  
    int a = 0;  
    for(int i = 0; i < n; i++)  
        for(int j = 0; j < n*n; j++)  
            a++;  
}
```

## **Oppgave 2.1 - 10%**

For hver av de tre programdelene navngitt alg1...alg3 skal du angi kompleksiteten (arbeidsmengden) i O-notasjon, inkludert en kort begrunnelse for hvordan du kom frem til svaret.

## **Oppgave 2.2 - 5%**

Skriv kort om hvorfor O-notasjon kan være et nyttig verktøy i forbindelse med evalueringen av en algoritme.

## **Oppgave 2.3 - 5%**

Dersom to algoritmer analyseres og forventes å gi samme kjøretid, hva kan man gjøre for å bestemme hvilken algoritme som er raskest?

## **Slutt del 2**

### **Del 3 – Totalt 40 %**

Mobiltelefonprodusenten Hakkepeil har bedt deg å skissere et systemet for å lagre informasjon om bekjente/kontakter i telefonens minne, heretter kalt kontakter. En kontakt post inneholder navn (fornavn og etternavn skilt med mellomrom), e-postadresse samt privat-,arbeid- og mobiltelefonnummer. Telefonen som Hakkepeil produserer kan lagre maksimalt 500 kontakter i minne.

Du skal i denne oppgaven diskutere et løsningsforslag som gir et system for å holde rede på lagrede kontakter slik at de kan gjenfinnes raskt basert på navn eller telefonnummer. Systemet skal ha funksjonalitet for å legge til og slette kontakt. Det skal være mulighet for å søke etter kontakt basert på navn eller deler av navn. Det skal være mulighet for søk etter kontakt basert på et gitt telefonnummer.

#### **Oppgave 3.1 – 25 %**

Beskriv av en passende datastruktur med stor vekt på begrunnelsen for de valg du tar.

#### **Oppgave 3.2 – 5 %**

Forklar hvordan innsetting og sletting av kontakter forekommer i systemforslaget ditt.

#### **Oppgave 3.3 – 5 %**

Forklar hvordan et søk etter en kontakt basert på navn eller deler av navn som søkenøkkel virker i ditt system.

#### **Oppgave 3.4 – 5 %**

Forklar hvordan et søk etter kontakt der et telefonnummer er søkenøkkel virker i ditt system. Det kan ikke søkes etter deler av et telefonnummer, hele telefonnummeret må være gitt.

**NB:** Gjør de antagelser og begrensninger som du synes er nødvendig samt dokumenter disse.

### **Slutt del 3**

## Del 4 – Totalt 20%

David A. Huffman utviklet i 1952 en metode for komprimering av data, kjent som Huffman koding. Denne oppgaven omhandler dette emnet.

### Oppgave 4.1 - 20%

Anta at en fil som skal komprimeres er lest og gir frekvenstabellen som gitt i tabell 4.1. På bakgrunn av denne tabellen bruk Huffman algoritmen til å lage et kodetre. Tegn kodetreet og forklar hvordan du kom frem til det. Fyll inn feltet "Kode" i tabellen med de nye kodene til tegnene ut ifra Huffman treet.

| Tegn | Frekvens | Kode |
|------|----------|------|
| A    | 4        |      |
| B    | 40       |      |
| C    | 6        |      |
| D    | 2        |      |
| E    | 45       |      |
| F    | 30       |      |
| G    | 1        |      |
| H    | 15       |      |
| I    | 11       |      |

Tabell 4.1

### Slutt del 4

Lykke til og god jul!

André