

## 2D1343, TENTAMEN I DATALOGI FÖR ELEKTRO

Onsdagen den 7 mars 2007 kl 14–19

Maxpoäng på tentamen är 50. Betygsgränser: 25 poäng ger trea, 35 ger fyra, 45 ger femma. Hjälpmedel: Ett A4-papper med dina egna handskrivna anteckningar på framsidan och baksidan och kursboken eller en algoritmbok och lexikon.

### *Teorifrågor (10p)*

Nedan finns fem frågor om algoritmer och datastrukturer. Varje korrekt svar ger en poäng. Om du dessutom motiverar ditt svar får du två poäng på frågan.

- a. Vilken komplexitet har följande program-kod?

Kod i Python:

```
def calc(inl):
    n = len(inl)
    i=1
    sum=0
    while i < n :
        sum=sum+inl[i]
        i=i*2
    return sum
```

Motsvarande kod i java:

```
int calc(int[] inl){
    int n=inl.length;
    int i=1;
    int sum=0;
    while(i<n){
        sum=sum+inl[i];
        i=i*2;
    }
    return sum;
}
```

- b. Var i en heapvektor finns pappan och sönerna till element nr 17?
- c. Kan man säga att djupetförstsökning är alltid effektivare än breddenförstsökning?
- d. Vilken funktion växer snabbast för stora  $n$ ,  $n^2$  eller  $n\log(n)$ ?
- e. Rekommenderar du binärsökning för att söka i en osorterad vektor?

### *Problemfrågor*

2. *Balanskonst*

På filen SAOL finns 120259 ordlistord i bokstavsordning. Man skulle vilja ha en funktion `readTree(n)` som vid anropet `readTree(120259)` returnerar en pekare till ett välbalanserat träd med ett ord i varje post. Den första tanke man får är att leta sig fram till det mittersta ordet i filen, skapa rotposten och stoppa in mittordet, leta reda på tjugofemprocents- och sjuttiofemprocentsorden och stoppa in i poster som man skapar under rotposten osv. Slå nu bort den tanken och försök i stället komma på en rekursiv tanke! Ingen snabbspolning hit och dit ska förekomma utan filen ska läsas igenom en enda gång. Den rekursiva tanken får uttryckas i ord, Python eller Java.

(8p)

3. *Bokningssyntax*

(6p)

Vi har en databas för ett bokningssystem. Databasen är på textformat. Första raden i databasfilen är ett tal som anger hur många minuter en redovisning tar. Sedan följer bokningshuvuden och bokningar. Varje bokningshuvud är en rad som inleds med tecknet > och följs av datum, starttid och sluttid för ett redovisningspass. Sist på raden kommer en sträng som anger i vilken sal redovisningarna ska äga rum och vem som ska ta hand om redovisningarna. Fälten skiljs av semikolon. Efter bokningshuvudet kommer de bokningar som hittills är gjorda till det redovisningstillfället, en bokning per rad. Varje bokningsrad inleds med tecknet : och följs av ordningsnumret, användarnamn samt efternamn och förnamn på den som har bokat in sig på tiden. Här skiljs också fälten av semikolon. Efternamn och förnamn skiljs av ett kommatecken. Följande är ett exempel:

```
20
>970129;8.10;10.00;gul;viggo
:2;sbyke;Byåker,Sölve
:3;keltor;Ekrot,Torkel
```

Ska tolkas som:

- Varje redovisning tar 20 minuter.
- Redovisningstillfället börjar klockan 8.10 och slutar klockan 10.00.
- Redovisningen sker i salen gul och den som tar redovisningen är viggo.
- klockan 8.30 ( $= 8.10 + (2 - 1) \times 0.20$ ) redovisar Sölve Byåker som har användarnamnet sbyke.
- Torkel Ektor, med användarnamnet keltor redovisar klockan 8.50 ( $= 8.10 + (3 - 1) \times 0.20$ ).

En fil med fler redovisningstillfällen kan se ut som nedan:

```
20
>070129;8.10;10.00;gul;viggo
:2;sbyke;Byåker,Sölve
:3;keltor;Ekrot,Torkel
>070130;8.05;23.00;orange;henrik
:1;golf;Holm-Ström,Gottwolf
:2;myskott;Mysk,Gottward
:6;lionn;Bonn,Lissa
:10;tennis;Tessin,Nisse
```

Din uppgift är att skriva en BNF för ovanstående bokningssystem som kan användas för att upptäcka syntax-fel på bokningsfiler.

4. *kackerlack-automat*

(7p)

Konstruera och rita upp en KMP-automat som söker efter insekten **harlekinkackerlacka** i en textfil. Ange next-vektorn och förklara hur man bygger next-vektorn. Visa hur din KMP-automat använder sig av next-vektorn när man vill söka efter en **harlekinkackerlacka** i textsträngen nedan:  
**harlekinmaushackeinkakerlakanhahahaharlekinkackerlackan**

5. *Bredden först*  
(6p) Man vill söka igenom ett binärträd bredden-först för att hitta ett visst personnummer. Kan någon av ordningarna preorder, inorder och postorder användas? Vilka av dem är odugliga? Varför? Förklara i ett litet exempel, nämligen ett balanserat träd med sju poster.
6. *Komprimering*  
(8p) Använd Huffmankodning för att komprimera följande mening:  
KALLE ANKA SATT PÅ EN PLANKA ROPADE SÅ HÅR, KVACK KVACK KVACK.  
Rita upp en Huffmanträd och förklara hur trädet byggs upp? Vad blir ordet ANKA huffmankodat?
7. *Abstrakta atomer*  
(5p) Varje atom har ett unikt nummer, beteckning och vikt. Man skulle kunna lagra information om atomer i var sin lista (en för nummer, en för beteckning och en för vikt) eller kanske lagra det som en abstrakt datatyp. Vilket av dem är bättre? Motivering krävs!

**Glöm inte klicka på kursvärderingsfrågor på kursens webbsida.**