

## 2D1320, TENTAMEN I TILLÄMPAD DATALOGI

Torsdagen den 26 augusti 2004 kl 14–19

Maxpoäng = 50. Betygsgränser: 25 poäng ger trea, 35 ger fyra, 45 ger femma. Resultatet anslås om ca två veckor på Nadas anslagstavla på Osquars backe 2, plan 3. Hjälpmedel : En algoritmbok och formelsamlingen.

### 1. *Sommarfoton*

Tilda har köpt en digitalkamera och tagit ca 20 foton varje dag under hela sommaren (juni, juli, augusti). Varje foto tar 1 MB. Det drar ihop sig till åskväder och hon vill spara alla bilder på CD omifall blixten skulle slå ner i datorn. Om hon komprimerar varje bild för sej går dom bara nästan in på en CD, men komprimering av alla som en jättefil blir tillräckligt effektivt.

- (5p) Om komprimeringsalgoritmen tar tid proportionell mot  $n^2$ , och en bild tar 2 millisekunder att komprimera, hur lång tid tar det att komprimera alla bilderna? En uppskattning av tiden räcker, men du bör kunna avgöra om det är rimligt att hinna med komprimeringen innan åskvädet drar fram!

### 2. *Söka rekursivt bland sommarbilderna*

Filnamnen på alla sommarbilderna finns nu i en array med index från 0 till  $n$ , sorterad i bokstavsordning. Vi vill söka efter index till filen med namnet "en riktig sommardag" i arrayen.

- (5p) Skriv en rekursiv tanke för att få fram index för en given fil i arrayen. För full poäng krävs en effektiv algoritm.

### 3. *Taltävling*

Under sommaren har det gått en tävling i radio där det gäller att så snabbt som möjligt hitta ett tal som uppfyller två villkor. Första villkoret, som gäller hela tävlingen, är att siffrorna i talet måste vara strikt stigande (23578 är OK men inte 235578 eller 23587). Andra villkoret är nytt för varje dag och kan t ex vara att talet ska vara ett primtal, jämnt delbart med 599 eller en tvåpotens.

- (8p) Beskriv utförligt en breddenförst- eller djupetförstsökning (effektivare metod ger fler poäng) som hittar det minsta talet som uppfyller villkoren. Vilka klasser och metoder behövs?

### 4. *Mergesort och quicksort*

Mergesort och quicksort är två snabba sorteringsalgoritmer. Ge tre konkreta exempel på problem där

- (6p)
- mergesort är att föredra framför quicksort
  - quicksort är att föredra framför mergesort
  - både mergesort och quicksort är olämpliga

5. *Teori*

(10p) Nedan finns fem frågor om algoritmer och datastrukturer. Varje fråga kan ge upp till två poäng. Motivering krävs!

- a. Kan man genomföra djupetförstökning med en kö istället för en stack?
- b. Går det att spara ett sorterat binärträd på fil för att senare kunna läsa in det igen?
- c. Om en ordlista ligger lagrad i en hashtabell, kan man därifrån skriva ut den i bokstavsordning?
- d. Finns det någon skillnad mellan en abstrakt datastruktur och en implementation av en datastruktur?
- e. Vilken funktion växer snabbast för stora  $n$ ,  $2^n$  eller  $n^2$ ?

6. *Hashade lösenord*

Vet du varför systemgruppen insisterar på att ge dig ett nytt lösenord istället för att slå upp ditt gamla när du glömt det efter sommaren? Det är för att lösenorden av säkerhetsskäl inte finns lagrade i klartext.

(5p) Beskriv hur ett lösenordssystem som använder hashning skulle kunna fungera. Vilka krav bör ställas på hashfunktionen och hashtabellen?

7. *Nytt kösystem*

När utskrifter skickas till skrivaren används ett kösystem som kallas FiFo (first in first out), som innebär att dokumenten skrivs ut i samma ordning som dom köas. En nackdel med detta system är att ett litet dokument kan få vänta länge om det hamnade i kön efter flera stora dokument.

(5p) I ett mer intelligent kösystem skulle ordningen i kön kunna bero både på dokumentets storlek och antal sekunder ett dokument har väntat på att skrivas ut, t ex enligt formeln  $Prioritetsnummer = DokumentetsVäntetid * 0.5 + DokumentetsStorlek$ . Din uppgift är att föreslå en datastruktur för en sådan kö och förklara hur den fungerar i detalj, samt att ange i vilken ordning följande dokument kommer att skrivas ut. Anta att skrivaren skriver ut 0.5 MB data/s och att kön ändras endast då något dokument läggs in eller tas ut ur kön.

mottagningsklockslag	dokumentets namn	dokumentets storlek
10.00.00	D1	3 MB
10.00.02	D2	5 MB
10.00.04	D3	2 MB
10.00.08	D4	3 MB
10.00.09	D5	1 MB

8. *Utskrivna binärträd*

(6p) Här är en utskrift av noderna i ett fullt, balanserat binärträd: A B C D E F G  
Hur ser trädet ut (rita!) om utskriften var i

- a. preorder
- b. inorder
- c. postorder