

Tentamen

Datastrukturer för D2

DAT 035

17 december 2005

- Tid: 8.30 - 12.30
- Ansvarig: Peter Dybjer, tel 7721035 eller 405836
- Max poäng på tentamen: 60. (Bonuspoäng från övningarna tillkommer.)
- Betygsgränser: 3 = 30 p, 4 = 40 p, 5 = 50 p
- Inga hjälpmedel.
- Skriv tydligt och disponera papperet på ett lämpligt sätt.
- Börja varje ny uppgift på nytt blad.
- Skriv endast på en sida av papperet.
- **Kom ihåg:** alla svar ska motiveras väl!
- Poängavdrag kan ges för onödigt långa, komplicerade eller ostrukturerade lösningar.
- Lycka till!

1. Vilka av följande påståenden är korrekta och vilka är felaktiga? Motivera! Svar utan bra motivering ger inga poäng.

- (a) Om man implementerar en stack med hjälp av ett dynamiskt fält (“extendable array”) så blir operationerna push (sätt in ett element överst i stacken) och pop (ta bort översta elementet i stacken) $O(1)$ i värsta fall.
- (b) Om man implementerar en stack med hjälp av en enkellänkad lista blir operationerna push och pop $O(1)$ i värsta fall.
- (c) Om ett binärt träd har n noder och höjden är h så är alltid $h \geq \log_2 n$. Ett träds höjd definieras som maximala noddjupet. Tänk på att roten ligger på djupet 0.
- (d) Om ett (2,4)-träd har n noder (och alltså lagrar maximalt $3n$ element) och höjden är h så gäller att $h \leq \log_2 n$.
- (e) Det är lämpligt att använda en hashtabell för att lagra svenska ord som används av ett program som gör rättstavningskontroll.

(10p)

2. Du ska lagra mängden $(7, A), (4, B), (1, C), (5, D)$ av par av nycklar och värden i fyra olika datastrukturer: heap, binärt sökträd, splayträd och skiplista. I alla fallen ska du använda standardalgoritmen för att sätta in nya element. Elementen ska sättas in ett efter ett i den ovan angivna ordningen.

Heap. Här ska du alltså använda heltalsnycklarna som prioriteter. Du ska börja med den tomma heapen och sedan visa steg för steg hur standardalgoritmen bygger upp heapen! Rita fyra bilder som visar hur heapen successivt byggs upp. Skriv också förklarande text så det framgår hur algoritmen fungerar.

Binärt sökträd. Här behöver du bara rita det slutgiltiga trädet. Sökträdet ska vara ordnat med avseende på nycklarnas storlek. Det ska inte balanseras.

Splayträd. Rita fyra bilder som visar hur splayträdet successivt byggs upp. Bifoga förklarande text.

Skiplista. Här behöver du bara rita den slutgiltiga skiplistan, som är ordnad m a p nycklarnas storlek. De fyra paren ska ha multipliciteterna 2,1,1,3 respektive. Dvs paret $(7, A)$ ska ha en kopia, paren $(4, B)$ och $(1, C)$ ska inte ha några kopior, och paret $(5, D)$ ska ha två kopior.

(10p)

3. (a) Vad gör metoden `f` nedan? Indatafältet `a` är sorterat och har längden n .

```
public int f(int k, int[] a) {
    return g(k,a,0,a.length-1);
}

private int g(int k, int[] a, int i, int j) {
    if (i > j) return -1;
    int m = (i+j)/2;
    if (k == a[m]) return m;
    if (k < a[m]) {return g(k,a,i,m-1);}
    return g(k,a,m+1,j);
}
```

- (b) Vilken asymptotisk tidskomplexitet har metoden `f` i värsta fallet? Motivera!
- (c) Vilken asymptotisk tidskomplexitet har den i bästa fallet? Motivera!
- (d) Vilken asymptotisk tidskomplexitet har följande Javametod?

```
public int h(int n) {
    int s = 0;
    for (i = 0; i < n; ++i) {
        for (j = i; j < n; ++j) {s += j;};
    }
    return s;
}
```

(10p)

4. (a) Lilla fågelboken med 100 uppslagsord och stora fågelboken med 10000 uppslagsord ligger lagrade i datorsökbar form. Det tar dubbelt så långt tid att söka i stora fågelboken som i lilla fågelboken. Vilken sökmetod har använts? Motivera.
- (b) Man bestämmer sig nu för att byta lagrings- och sökmetod. Nu tar det lika långt tid att söka i båda fågelböckerna. Vilken är den nya sökmetoden? Motivera.

(5p)

5. Antag att du har en graf där noderna representerar städer och bågarna representerar vägar mellan städerna och hur långa dessa vägar är. Handelsresandeproblemet ("the Travelling Salesperson Problem") består i att finna den kortaste rundturen som besöker alla noder. En rundtur är en cykel, dvs en väg med samma start- och slutnod.

(a) Betrakta följande algoritm. Börja i en godtycklig startnod. Gå sedan till den närmaste grannen. Gå därefter till denne grannes närmaste obesökta granne. Osv.

Ger algoritmen en korrekt lösning till handelsresandeproblemet? Om du svarar ja måste du motivera ditt svar. Om du svarar nej ska du ge ett motexempel.

(b) Det finns ett antal välkända algoritmdesignmetoder som togs upp i kursen. Vilken metod är ovanstående algoritm ett exempel på? Motivera.

(5p)

6. (a) Du vill sortera telefonnummerna i Göteborgskatalogen i lexikografisk ordning. Nummerna är från början helt osorterade. Antag att alla telefonnummer har 6 eller 7 siffror. Vi betraktar ett 6-siffrigt nummer som ett 7-siffrigt nummer med ett sista blanktecken. Lexikografisk ordning innebär att man i första hand sorterar efter första siffran i numret, i andra hand efter andra siffran, osv. Blanktecknet betraktas som mindre än varje annan siffra. Här är ett exempel på korrekt sorterade telefonnummer.

405837
4058372
405839
406100
4061000
511099

Du ska ange två sorteringsalgoritmer som du betraktar som huvudkandidater för uppgiften och jämföra dem. Du måste motivera ditt svar med utgångspunkt från algoritmernas asymptotiska komplexiteter och att antalet telefonnummer som ska sorteras är ca 300 000.

(b) Efter det att du har sorterat katalogen ska du sätta in 1000 nummer till. Vilken sorteringsalgoritm är mest effektiv för denna uppgift? Även här ska du motivera ditt svar utifrån asymptotiska komplexiteter.

(c) Ange två datastrukturer som är lämpliga för att lagra Göteborgskatalogen i och jämför dem. Även här ska du motivera ditt svar genom att ange asymptotiska komplexiteter för sökning, insättning och borttagning.

(10p)

7. Skriv en algoritm som löser Sudoku-problem!

Ett Sudoku-problem består av en 9×9 - matris där en del av cellerna är ifyllda med siffrorna 1 - 9. Att lösa ett Sudoku-problem innebär att fylla i de tomma rutorna så att

- varje rad innehåller alla siffrorna 1-9 precis en gång.
- varje kolumn innehåller alla siffrorna 1-9 precis en gång.
- var och en av de nio 3×3 -delmatriserna (som uppstår om man delar 9×9 -matrisen i 3×3 delar, se figur) innehåller alla siffrorna 1-9 precis en gång.

Du kan utgå från att det givna Sudoku-problemet har en entydig lösning. Här är ett exempel på ett Sudoku-problem.

```
- - 8 5 7 3 - 9 -  
- - 5 - - - - - 7  
- 2 6 - 4 9 - 8 -  
  
- - - 7 1 8 - - -  
- - - - - - - - -  
- - - 9 6 5 - - -  
  
- 7 - 3 9 - 6 1 -  
3 - - - - - 4 - -  
- 1 - 6 2 7 5 - -
```

Du ska använda maximalt en sida (med normalstor text) för att beskriva en algoritm som löser detta problem. Din lösning bör tala om vilka datastrukturer och algoritmer du använder, och hur de implementeras. Om du vill kan du använda pseudokod på "hög nivå". Du kan förutsätta att algoritmer och datastrukturer från kursen är kända.

Poängsättningen kommer att grundas på hur klar och fullständig beskrivning du ger. Det är också viktigt att du väljer de mest lämpliga datastrukturerna och algoritmerna, så att problemet kan lösas på ett effektivt sätt. (10p)