

Omtentamen - Datastrukturer, algoritmer och programkonstruktion.

DVA104

Akademien för innovation, design och teknik

Torsdag 2014-08-14

Skrivtid: 14.30-19.30
Hjälpmedel: Inga
Lärare: Caroline Uppsäll, 021-101456

Betygsgränser

Betyg 3: 27p - varav minst 8 poäng är P-uppgifter
Betyg 4: 40p
Betyg 5: 48p
Max: 53p - varav 16 poäng är P-uppgifter

Allmänt

- Skriv endast en uppgift per blad
- Skriv bara på ena sidan av pappret.
- Referera inte mellan olika svar.
- Om du är osäker på vad som avses i någon fråga, skriv då vad du gör för antagande.
- Oläsliga/Oförståeliga svar rättas inte.
- Kommentera din kod!
- Eventuellt intjänade bonuspoäng kan/kunde endast användas på den ordinarie tentamen. Då detta är en omtentamen kan alltså inte några bonuspoäng räknas med.
- Tips: Läs igenom hela tentan innan du börjar skriva för att veta hur du ska disponera din tid.

Lycka till!

Uppgift 0: (0p)

Läs noga igenom instruktionerna på förstasidan och följ dem!

Uppgift 1: (8p)

- a) Vad menas med divide & conquer? Förklara principerna bakom begreppet och nämn några exempel på sådana algoritmer. (2p)
- b) Vad innebär det att ett binärt träd är fullt? (1p)
- c) Varför är basfallet viktigt i rekursion? (1p)
- d) Vad innebär det att en algoritm är stabil? (1p)
- e) Vad är ett annat namn på en LIFO-kö? (1p)
- f) Vad innebär det att ett träd är fullt? (1p)
- g) Nämn minst två förutsättningar som måste uppfyllas för att man ska kunna utföra binärsökning på en mängd. (1p)

Uppgift 2: (7p)

- a) Vad är det som gör att följande funktion är rekursiv? Nämn minst tre saker, ge exempel från koden och glöm inte att förklara. (3p)

```
int funk(int n)
{
    if (n < 0)
        return -1;
    else
    {
        if (n == 0 || n == 1)
            return 1;
        else
            return n * funk(n-1);
    }
}
```

- b) Vad gör funktionen ovan? (1p)
- c) Skriv om den rekursiva funktionen till en iterativ funktion (3p) ---**P-uppgift**

Uppgift 3: (5p)

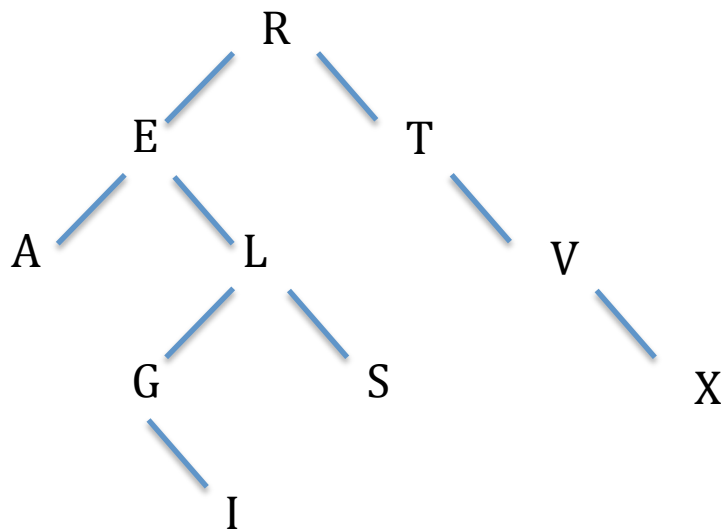
- Skapa en arraybaserad cirkulär kö med plats för 10 element. Glöm inte att kommentera koden! (2p) --- **P-uppgift**
- Skriv funktionen för att ta bort element i din cirkulära kö (köns storlek får inte ändras). Glöm inte att kommentera koden! (3p) --- **P-uppgift**

Uppgift 4: (9p)

- Beskriv hur följande algoritmer fungerar: (3p)
 - Bubbelsortering (Bubblesort) – ej optimerad
 - Insättningssortering (Insertionsort)
 - Urvalssortering (Selectionsort)
- Jämför ovanstående tre algoritmer med avseende på komplexitet (komplexitetsklass) i bästa samt värsta fall. Diskutera när de olika algoritmerna är bra respektive mindre bra att använda (och varför). (4p)
- Nämn två sätt man kan optimera standardlösningen av bubbelsortering på. (2p)

Uppgift 5: (10p)

Antag följande träd:



- Är följande träd ett binärt sökträd eller inte? – Motivera! (1p)
- Skriv ut trädets preorder. (1p)
- Är trädets balanserat? (1p)
- Visa hur trädets utseende ser ut efter datat 'K' har lagts till i trädets. Anser du att trädets ovan inte är ett balanserat binärt sökträd så ritar du först om det så att det blir ett sådant. (1p)

- e) Skriv koden för noden som används i trädet ovan. (1p) ---**P-uppgift**
- f) Skriv funktionen som lägger till data i trädet. Funktionen ska vara rekursiv och ta som minst det nya datat (inte den nya noden) som inparameter. Glöm inte att kommentera koden. (3p) ---**P-uppgift**
- g) Förklara varför man bör använda rekursion när man ska ta reda på hur djupt/högt ett binärt träd är. (2p)

Uppgift 6: (4p)

Ta en från början tom hashtabell som löser krockar med hjälp av öppen adressering (linear probing) med 10 platser vars hashfunktion är $x \bmod 10$.

- a) Rita en bild över vad som händer när följande sekvens av nycklar sätts in: (2p)
35, 2, 18, 6, 3, 10, 8, 5
- b) Rita en bild över hur hashtabellen skulle se ut om den istället var länkad (samma data som i a ska sättas in i tabellen) (2p)

Uppgift 7: (6p)

- a) Visa hur arrayen nedan sorteras steg för steg med Quicksort. Du väljer själv vilket tal som ska väljas som pivot (se dock till att vara konsekvent). (4p)

6	1	7	3	5	0	2	8	4
---	---	---	---	---	---	---	---	---

- b) Diskutera hur valet av pivotvärde kan påverka algoritmens komplexitet. Rita gärna bilder för att komplettera din diskussion. (2p)

Uppgift 8: (4p)

Du ska nu skriva **pseudokoden eller koden** (du väljer själv, skriv dock en kommentar om vad du valt) för den övergripande funktionen (MergeSort) i en Mergesort-algoritm. Du kan anta att mängden som ska sorteras är en array av heltal och att du till ditt förfogande har en funktion med funktionshuvudet

`Merge(integer first, integer mid, integer last)`

som kombinerar ihop delarrayer till en ny sorterad array/delarray. Funktionen Merge ska givetvis anropas på vettig plats i din pseudokod. --- **P-Uppgift**