Tentamen - Datastrukturer, algoritmer och programkonstruktion.

DVA104

Akademin för innovation, design och teknik

Måndag 2015-01-12

Skrivtid: 14.30-19.30
Hjälpmedel: Inga
Lärare: Caroline Uppsäll, 070-4616110

Betygsgränser

Betyg 3:	21p - varav minst 7 poäng är P-uppgifter
Betyg 4:	29p
Betyg 5:	35p
Max:	39p - varav 13 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 endast användas på ordinarie tentamenstillfälle.
- 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: *(6p)*

a) Varför är basfallet viktigt i rekursion? *(1p)*
b) Vad innebär det att en algoritm är stabil? *(1p)*
c) Vad är ett annat namn på en stack? *(1p)*
d) Vad innebär det att en algoritm har konstant komplexitet? *(1p)*
e) Nämn två sätt på vilka man kan optimera standardlösningen av en bubblesort. *(2p)*

Uppgift 2: *(6p)*

a) Beskriv hur binärsökning i en linjär sekvens går till. *(1p)*
b) Skriv koden för en funktion som utför binärsökning på en sorterad linjär sekvens. Du kan anta att sekvensen är en sorterad array av heltal. Arrayen och det eftersökta värdet ska tas som inparametrar till funktionen. *(3p)* ***---P-uppgift***
c) Vilken komplexitet har funktionen och varför? *(2p)*

Uppgift 3: *(11p)*

a) Rita det sorterade binära trädet där heltalen 25 14 38 43 9 39 22 15 37 sätts in i just den ordningen. *(1p)*

b) Skriv ut trädet från uppgift a i preorder? *(1p)*

c) Är trädet balanserat? *(1p)*

d) Skriv koden för noden som används i ett sorterat binärt träd som håller heltal. (2p) ***---P-uppgift***

e) 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. *(4p)* ***---P-uppgift***

f) Förklara varför man bör använda rekursion när man ska ta reda på hur djupt/högt ett binärt träd är. *(2p)*

Uppgift 4: *(6p)*

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

a) Rita en bild över vad som händer när följande sekvens av nycklar sätts in: *(2p)*
32, 5, 12, 8, 3, 11, 30, 41

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)*

c) Har man en väldigt stor hashtabell till en mängd som med största säkerhet inte överskrider hashtabellens storlek så ökar man avsevärt chanserna att få en komplexitet närmare O(1) på sökning i tabellen. Finns det någon negativ aspekt av detta som bör tas i beaktning? – Motivera/Diskutera *(2p)*

Uppgift 5: *(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 6: *(4p)*

Algoritmen MergeSort består av två funktioner. Den övergripande funktionen delar upp mängden och anropar sedan den andra funktionen som kombinerar ihop delmängderna till en ny sorterad delmängd.

Din uppgift är att skriva **pseudokoden** för den övergripande funktionen, som ska heta `MergeSort`. Du kan anta att mängden som ska sorteras är en array av heltal, till ditt förfogande har du funktionen som kombinerar delmängderna, dess funktionshuvud ser ut såhär:

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

Funktionen `Merge` ska givetvis anropas på vettig plats i din pseudokod. --- ***P-Uppgift***