

Tentamen - Datastrukturer, algoritmer och programkonstruktion.

DVA104

Akademien för innovation, design och teknik

Måndag 2015-08-17

Skrivtid: 14.30-19.30
Hjälpmedel: Bok relaterad till ämnet – ej digital (inga föreläsningsanteckningar eller egna anteckningar är tillåtna)
Lärare: Caroline Uppsäll, 021-101456

Betygsgränser

Betyg 3:	25p - varav minst 8 poäng är P-uppgifter
Betyg 4:	36p
Betyg 5:	42p
Max:	47p - varav 14 poäng är P-uppgifter

Allmänt

- Kod skriven i tentauppgifterna är skriven i pseudokod.
- På uppgifter där du ska skriva kod (P-uppgifter) ska koden skrivas i det språk som var aktuellt då du tog kursen (C eller Ada). Ange högst upp på bladet vilket språk koden är skriven i.
- 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: (9p)

- a) Varför är basfallet viktigt i rekursion? (1p)
- b) Vad innebär det att en sorteringsalgorithm är stabil? (1p)
- c) Vad innebär det att en algorithm har kvadratisk komplexitet? (1p)
- d) Nämn två sätt på vilka man kan optimera standardlösningen av en bubblesort. (2p)
- e) Beskriv hur en stack (LIFO-kö) fungerar. (1p)
- f) Vad används ordobegreppet (O-notation) till? (1p)
- g) Beskriv kortfattat vad en dubbellänkad lista är och hur en sådan fungerar. (2p)

Uppgift 2: (6p)

- a) Beskriv hur binärsökning i en linjär sekvens går till, finns det några förutsättningar som måste vara uppfyllda för att binärsökning ska kunna tillämpas? (1p)
- b) Skriv koden för en funktion som utför binärsökning på en sorterad linjär sekvens. Du kan anta att sekvensen är en array/ett fält av heltal. Sekvensen och det eftersökta värdet ska tas som inparametrar till funktionen. (3p) **---P-uppgift**
- c) Vilken komplexitet har funktionen (i bästa samt sämsta fall) och varför? (2p)

Uppgift 3: (5p)

Antag följande rekursiva funktion (skriven i pseudokod)

```
Integer Funk(Integer i, Integer x)
    If x <= 10 AND i > 0 then
        Return 2 + Funk(i-1, x*2)
    Else then
        Return i
End Funk
```

- a) Beskriv vilket resultatet blir efter följande anrop (vad ligger i variabeln T efter att anropet har körts) och motivera varför (med bild och/eller text). (3p)
 - 1) Integer T = Funk(5, 1)
 - 2) Integer T = Funk(4, 5)
 - 3) Integer T = Funk(0, 0)
- b) Skriv om ovanstående rekursiva funktion till en iterativ algorithm som gör samma sak. (2p) **---P-uppgift**

Uppgift 4: (8p)

- a) Vad innebär det att ett binärt träd är balanserat? (1p)
- b) Rita upp ett binärt sökträd med elementen i nedanstående lista. Ta elementen ett och ett från vänster till höger ur listan nedan och lägg in i trädet. (2p)
100, 50, 75, 60, 200, 150, 180, 40, 90, 210
- c) Vilken traverseringsmetod bör användas på ditt träd ovan (från uppgift b) för att skriva ut elementen i storleksordning (från minsta till största)? (1p)
- d) Visa hur utskriften av ditt träd ovan (från uppgift b) ser ut om traverseringsmetoden postorder används. (1p)
- e) Antag att du har noden nedan (skriven i pseudokod). Skriv funktionen som letar upp ett angivet data i trädet. Du väljer själv om funktionen ska vara rekursiv eller iterativ. Det eftersökta värdet ska tas som parameter till funktionen och 1/true eller 0/false ska returneras beroende på om datat hittats i trädet eller inte. Du kan anta att det inte finns några dubletter i trädet samt att alla "tomma" länkar är satta till NULL. Länkarna (leftChild och rightChild) är pekare/referenser. (3p) ---**P-uppgift**

```
BTreeNode  
    Integer Node  
    BTreeNode leftChild  
    BTreeNode rightChild
```

Uppgift 5: (7p)

- a) Ta en från början tom listad (chained) hashtabell vars hashfunktion är $x \bmod 8$. Hashtabellen har 8 platser. Rita en bild över vad som händer när följande sekvens av data sätts in (i ordning från vänster till höger). (2p)
48, 72, 98, 66, 84, 18, 34, 86, 8, 4, 12, 94
- b) Bedöm hashfunktionens kvalitet, diskutera med avseende på t.ex. minnesåtgång, komplexitet etc. (3p)
- c) Beskriv hur hashning med öppen adressering (linear probing) fungerar, vad händer om ovanstående sekvens av data läggs till i en hashtabell som använder just öppen adressering för att hantera krockar? (2p)

Uppgift 6: (6p)

- a) Visa med en bild hur följande sekvens sorteras med hjälp av MergeSort. (2p)

48	72	98	66	84	18	34	86	8	4	12	94
----	----	----	----	----	----	----	----	---	---	----	----

- b) Gör en klassificering av MergeSorts egenskaper (bästa komplexitet, sämsta komplexitet, minnesåtgång, naturlig, stabil). Glöm inte att förklara klassificeringen. (4p)

Uppgift 7: (3p)

Definiera en ADT för en kö (FIFO). Informationsdelen ska vara ett heltal. Glöm inte pre- och postconditions samt eventuellt returvärde. Kön ska använda sig av nedanstående länkade lista (skriven i pseudokod). Länkarna (next, previous, head och tail) är pekare/referenser. (3p) **---P-uppgift**

```
LinkedListNode
    Integer data
    LinkedListNode next
    LinkedListNode previous
```

```
LinkedList
    LinkedListNode head
    LinkedListNode tail
```

Uppgift 8: (3p)

Sorteringsfunktionen QuickSort använder sig av två funktioner. En (QuickSort) som hanterar den övergripande logiken och en (Partition) som plockar ut pivot-värde samt placerar mindre värden till vänster och större till höger. Den senare har följande funktionshuvud (skrivet i pseudokod) där sekvensen är en array/ett fält av heltal.

```
Integer Partition(Integer sekvens, Integer first, Integer last)
```

Din uppgift är att skriva pseudokoden (eller koden, du väljer själv) för den övergripande, rekursiva funktionen QuickSort. Visa också hur det initiala anropet till funktionen ser ut. (3p) **---P-uppgift**