Tentamen i Algoritmer och Datastrukturer DV1 1998-02-20

Skrivtid: 1500-2100
Hjälpmedel: Räknedosa, *Några användbara matematiska samband*
Varje delfråga kan, om inget annat anges, ge högst 1 poäng.
Svar skall motiveras om inte annat anges.
Lycka till!

Uppgifter

1. Nedan följer ett antal påståenden. Ange, utan motivering, för var och ett av dessa huruvida det *alltid* är korrekt eller ej. Rätt svar ger 0.5 poäng, fel svar ger -1.0 poäng och uteblivet svar ger 0 poäng. Totalpoängen på uppgiften kan dock aldrig bli negativ.
   (a) $\log(n^3) + n^2$ är $O(n^2)$
   (b) $n + n^2$ är $\Omega(n)$
   (c) $n + n^2$ är $\Omega(n^2)$
   (d) Tiden för quicksort är $O(n \log n)$
   (e) Att söka en nyckel i ett AVL-träd kräver $O(\log n)$ operationer.
   (f) För tillräckligt stora mängder är quicksort (med en vettig implementation) alltid snabbare än enkel instickssortering.

2. Mätningar av tiderna för en sorteringsmetod har givit följande värden

Antal element	Tid (ms)
1000	1.0
2000	3.8
4000	15.7

   (a) Implementatören påstår att detta är en $O(n \log n)$-metod. Verkar det stämma med värdena? Om det inte stämmer, ange vilket tidsberoende metoden verkar ha. Motivera!
   (b) Hur lång tid kan man förvänta sig att det tar att sortera 1000000 element med denna metod?

3. Beskriv algoritmen för *binär sökning* av sorterade nycklar lagrade i en array. Visa att algoritmen är $O(\log n)$ (2p)

4. (a) Definiera en *heap* (i samband med prioritetsköer och heapsortering)!
   (b) Beskriv en algoritm för att konstruera en heap med n element som tar $O(n)$ tid.
   (c) Visa att algoritmen i (b) tar $O(n)$ tid. (2p)

5. (a) Vad menas med Fibonacci-träd?
   (b) Visa att höjden i ett Fibonacci-träd är $O(\log n)$

6. Beskriv med ord och figur hur en dubbelrotation i samband med inlägg i AVL-träd går till. Ur beskrivningen skall det framgå att rotationen säkert återställer AVL-egenskapen. (Alternativt kan motsvarande göras för inlägg i ett röd-svart träd.) (2p)

7. Följande nycklar är givna: 3 12 7 2 19 5 23 20 15.
   (a) Använd dessa, i given ordning, för att bygga upp en hashtabell med *dubbel hashning.* Hashtabellen skall ha 13 platser.
   (b) Använd nycklarna, i given ordning, för att bygga upp en *ordnad* hashhabell med *dubbel hashning* Hashtabellen skall ha 13 platser.
   (c) Vad menas med *likformig* (eng. *uniform*) hashning? När använder man den?

8. Följande funktion beräknar värdet av det n:te Chebyshevpolynomet i en viss punkt x.

```
Chebyshev(n,x) =
  if      n=0 then 1
  else if n=1 then x
  else             2*x*Chebyshev(n-1,x) - Chebyshev(n-2,x)
```

Hur beror tiden för denna funktion av n? (2p)