Tentamen i Algoritmer och Datastrukturer DV1 1999-03-19

Skrivtid: 0900 – 1500
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. Ett binärt sökträd skall representeras med hjälp av följande deklarationer

```
typedef struct treeNode {
  int item;
  struct treeNode *left, *right;
} treeNode;

typedef treeNode *link;

link cons( int it, link left, link right ) {
  link l;
  l = (link) malloc( sizeof(treeNode) );
  l->item  = it;
  l->left  = left;
  l->right = right;
  return l;
}
```

   (a) Skriv en C-funktion `int height(link r)` som beräknar och returnerar höjden av det träd som har sin rot i den nod som `r` pekar till.

   (b) Skriv en C-funktion `void insert(int key, link *r)` som lägger in `key` i det träd som har rot i den nod som `*r` pekar till. Exempel på användning:

```
link root = 0;      /* Pekare till ett tomt träd */
insert( 4, &root ); /* Lägg in 4 i det tomma trädet */
```

   (2p)

2. Följande nycklar är givna: 8 13 28 2 22 5 19 21 15 4.

   (a) Använd dessa, i given ordning, för att bygga upp en hashtabell med *dubbel hashning*. Hashtabellen skall ha 13 platser. Ange hur många försök det i genomsnitt krävs för sökning av existerande nyckel i den erhållna tabellen. (2p)

   (b) Använd nycklarna, i given ordning, för att bygga upp en *ordnad* hashhabell med *dubbel hashning*. Hashtabellen skall ha 13 platser. Vad är fördelarna med ordnad hashning framför vanlig dubbel hashning? (2p)

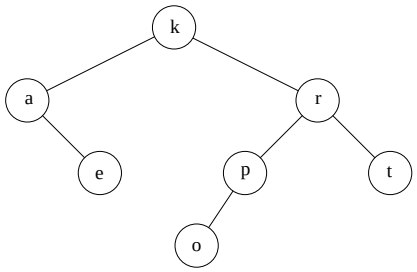
3. Antag att ett problems storleksordning beror på ett heltal n samt att det med ett visst program tar 1 sekund att lösa ett problem med $n = 1000$. Hur lång tid skulle det ta att lösa ett problem med $n = 1000000$ om tiden för programmet är

   (a) $O(1)$
   (b) $\Theta(\log n)$
   (c) $\Theta(n)$
   (d) $\Theta(n \log n)$
   (e) $\Theta(n^2)$

   Motivera svaren! Varje deluppgift ger maximalt 0.4p.

4. (a) Definiera begreppen *intern* och *extern väglängd* i binära träd.
   (b) Visa att $E = I+2n+1$ där E är den externa, I den interna väglängden och n antalet noder i trädet.
   (c) Vad används väglängdsbegreppen till i samband med binära sökträd?
   (d) Härled ett uttryck för hur den interna väglängden beror på antalet noder i värsta fall.
   (e) Härled ett uttryck för hur den interna väglängden beror av antalet noder i bästa fall. Det räcker här med att betrakta kompletta träd dvs träd där alla nivåer är helt fyllda. (2p)

5. (a) Visa med hjälp av ett exempel att ett rödsvart träd *inte* behöver vara ett AVL-träd. Du måste visa att det träd du konstruerat verkligen är ett rödsvart träd!
   (b) Givet följande AVL-träd med bokstäver i bokstavsordning som nycklar:



   Lägg in m och q så att trädet fortfarande förblir ett AVL-träd. Redovisa varje inlägg för sig. Om du använder andra algoritmer än standardalgoritmerna så måste du redovisa dessa. Alternativt kan du behandla trädet som ett rödsvart träd. (2p)

6. I ett problem skall *medianen* för stora mängder *heltal* i intervallet $[1, 1000]$ beräknas. Medianen av n tal är det tal som kommer på plats $n/2$ (heltalsdivision) om man lagrar talen i en array och sorterar dem. Ge en algoritm som tar *garanterat* $O(n)$ tid (n är antalet tal). Visa att algoritmen uppfyller kravet. (2p)