

SI-möte #7, Programkonstruktion och Datastrukturer

Elias Castegren & Kristiina Ausmees

elca7381@student.uu.se || krau6498@student.uu.se

16 December 2011

Begrepp

- i)* Vad är ett självbalanserade träd? Varför kan man vilja använda ett sådant?
- ii)* Vad innebär det att rotera ett (binärt) träd?
- iii)* Vad är invarianten i ett Red-Black tree? Hur upprätthålls den invarianten när man sätter in nya noder? Vad händer med nodernas färg när man roterar?
- iv)* Vad är invarianten i ett AVL-träd? Hur upprätthålls invarianten vid insättning? Vad menas med en nods balansfaktor?

Övningar

1.

Sätt in följande element (i ordning) i ett Red-Black tree:

4, 2, 8, 1, 3, 0, 7, 9, 5, 6

Kontrollera att invarianten är uppfylld efter varje insättning!

2.

Sätt in följande element (i given ordning) i ett tomt AVL-träd:

5, 3, 8, 2, 4, 1, 7, 9, 6

Rita upp hur hela trädet ser ut efter varje insättning. Skriv också ut varje nods balansfaktor.

(Fortsätt gärna att sätta in element om du vill öva mer på Red-black trees eller AVL-träd)

3.

Följande två funktioner beräknar produkten av två positiva heltal a och b :

```
fun swedish(0,b) = 0
  | swedish(a,b) = b + swedish(a-1,b);

fun russian(1,b) = b
  | russian(a,b) =
    if a mod 2 = 1 then
      b + russian(a div 2, b*2)
    else
      russian(a div 2, b*2);
```

Vad har funktionerna för tidskomplexitet? Skriv en svansrekursiv version av `russian`. Har den nya funktionen en bättre tidskomplexitet? Har den en bättre minneskomplexitet?

4.

Beräkna tidskomplexiteten i värsta fall för nedanstående två funktioner som använder *balanserade* binära sökträd:

```
datatype 'a tree = Leaf | Node of int * 'a * 'a tree * 'a tree

fun dataFromKey(_, Leaf) = NONE
  | dataFromKey(k, Node(k', d, L, R)) =
    if k > k' then
      dataFromKey(k, R)
    else if k < k' then
      dataFromKey(k, L)
    else
      SOME d;

fun keyFromData(d, Leaf) = NONE
  | keyFromData(d, Node(k, d', L, R)) =
    if d = d' then
      SOME k
    else
      let
        val k' = keyFromData(d, L)
      in
        if isSome k' then
          k'
        else
          keyFromData(d, R)
      end;
```

Lycka till!