

Algorithms and Datastructures assignment 3

Thomas Broby Nielsen (xlq119)

Tobias Overgaard (vqg954)

Christian Buchter (zvc154)

28. maj 2015

Indhold

1	Task 1	2
2	task 3	3
3	Task 4	4
4	Task 5	5

1 Task 1

```
get-kth-key(x,k)
1 if k < 0 or k > x.max
2   return NIL
3 else
4     if k > x.left.size
5         get-kth-key(x.right,k-x.left.size)
6     if k < x.left.size
7         get-kth-key(x.left,k)
8     if k = x.left.size
9         return  x.left.size+1
```

2 task 3

```
sized-Left-Rotate(T,x)
1 y = x.right
2 x.right = y.right
3 if y.right != T.nil
4     y.right.p = x
5 x.right.left = y.left
6 if y.left != T.nil
7     y.left.p = x.right
```

everything from here on is normal.

This way the size of x will not change when rotated.

3 Task 4

RB-INSERT(T, z)

1 y = T.nil

2 x = T.root

3 while x != T.nil

4 y = x

5 if z.key < x.key

6 x.size = x.size + 1

7 x = x.left

8 else x.size = x.size + 1

9 x = x.right

everything from here on is normal.

1

¹Se side 315 på "RB-Insert(T,z)"pseudo code

4 Task 5

Siden at vi kun har tilføjet to linjer kode, til den originale RB-INSERT, som arbejder i konstant tid, på linje 6 med indførelsen af " $x.size = x.size + 1$ " og igen på linje 8 med " $x.size = x.size + 1$ ", vil det ikke påvirke algoritmens originale køretid på $O(\lg n)$, som står skrevet i bogen².

Ændringerne som vi har lavet til den originale version af RB_delete, ændrer ikke køretiden. Dette er fordi at vores ændringerne kun ændrer size når den aligevel var i noden.

Query/get-kth-key kører i $O(\lg n)$ tid, da den går et trin ned i træet hver gang den bliver kaldt. Der er $\lg(n)$ niveauer.

²se side 322 "Analysis"