

# Algorithms and Datastructures assignment 3

Thomas Broby Nielsen (xlq119)

Tobias Overgaard (vqg954)

Christian Buchter (zvc154)

28. maj 2015

## Indhold

<b>1</b>	<b>Task 1</b>	<b>2</b>
<b>2</b>	<b>task 3</b>	<b>3</b>
<b>3</b>	<b>Task 4</b>	<b>4</b>
<b>4</b>	<b>Task 5</b>	<b>5</b>

# 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 rom here on is normal.

## 4 Task 5

RB\_insert Da ændringerne på algoritmen ikke påvirker køretiden er køretiden stadig  $O(\lg n)$ .

RB\_delete stadig i same tid, da ændringerne her ville kunne implementeres så de ikke påvirkede køretiden, men at den bare ændrede 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.