



```
Search(key,root):  
    if root = NULL or root.key = key  
        return root  
  
    if key < root.key  
        return Search(key,root.left)  
    else  
        return Search(key,root.right)
```

```
Insert(key, val, root):  
    if root = NULL  
        x = new node(key, val)  
        return x  
  
    if key < root.key  
        root.left = Insert(key, val, root.left)  
        root.left.parent = root  
    else  
        root.right = Insert(key, val, root.right)  
        root.right.parent = root  
  
    return root
```

```

AVL-Insert(key, val, root):
    if root = NULL
        x = new node(key, val)
***    x.height = 0
        return x

    if key < root.key
        root.left = Insert(key, val, root.left)
        root.left.parent = root
    else
        root.right = Insert(key, val, root.right)
        root.right.parent = root

*** root.height = 1+max(root.left.height, root.right.height)
*** return Rebalance(root)

```

Rebalance(x):

```
if x.left.height = x.right.height - 2  
    return LeftRotate(x)
```

```
else if x.right.height = x.left.height - 2  
    return RightRotate(x)
```

```
else return x
```

Je toto dobre?

Rebalance(x):

```
if x.left.height = x.right.height - 2
```

```
    y = x.right
```

```
    if y.right.height = y.height - 2
```

```
        y = RightRotate(y); y.parent = x; x.right = y
```

```
    return LeftRotate(x)
```

```
else if x.right.height = x.left.height - 2
```

```
    y = x.left
```

```
    if y.left.height = y.height - 2
```

```
        y = LeftRotate(y); y.parent = x; x.left = y
```

```
    return RightRotate(x)
```

```
else return x
```