Mohamed Khelif
Homework 4

13.3 A

Consider F_h represents the number of nodes with height h and F_h means the Hth Fibonacci number a number and as in an an AVL tree, a relationship exists that in an AVL tree the heights of two children subtree of any node differ by at most one. In an AVL tree the root has two children: one with height h-1 and the other with height at least h-2

$F_h = F_{h-1} + F_{h-2} +1$

$F_{h-1} = F_{h-2} + F_{h-3} + 1$

$F_h = (F_{h-2} + F_{h-3} + 1) + F_{h-2}+1$

$F_h > 2F_{h-2}$

$log(F_h) > log2^h/2$

2 loGh>h

$h=O(log(F_h)$

If $F_h == n$

Height h is O(logn)

B

```
BALANCE(x)
if(height(x.left)-height(x.right)<=1)
Return x
Else
if(height(x.left)<height(x.right))
Y = x.right
if(y.left<y.right)
Return left-rotate(x)
Else
Return left-rotate(x)
Else
Y = x.right
if(y.right<y.left)
Return right-rotate(x)
Else left-rotate(y)
Return right-rotate(x)
```

C

```
AVL-INSERT(x,z)
if(x==null)
Z.height = 0
Return z
if(z.key <= x.key)
Y = AVL-INSERT(x.left,x)
```

X.left = y
Else
Y = AVL-INSERT(x.right,x)
X.right = y
Y.parent = x
X.height = y.height+1
x= BALANCE(x)
Return x

D
It is already known that getting the height of an AVL Tree takes O(logn) time so the insertion and update operation will take time that is equal to O(logn). As seen in part b the height of an unbalanced tree decreases by 1 after a rotation. So the rotation process will take O(1) time. THerefore AVL-INSERT will take O(logn) time and rotation will take O(1)