

```

/* *
* Title : Binary Search Trees
* Author : ABDUL RAZAK DAHER KHATIB
* ID : 21801340
* Section : 001
* Assignment : 2
* Description : PDF FILE
*/

```

Q1)

a)

Preorder:

* - 13 ^ 3 2 + 4 / 21 7

Inorder:

13 - 3 ^ 2 * 4 + 21 / 7

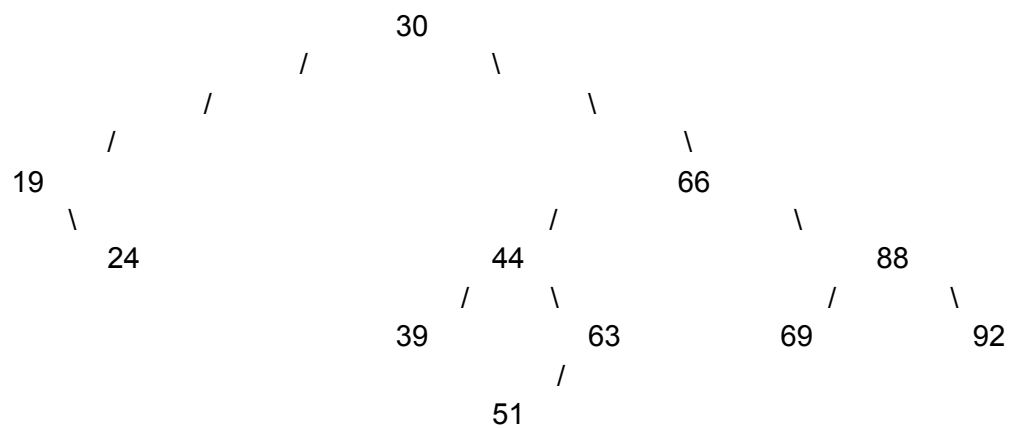
Postorder:

13 3 2 ^ - * 4 21 7 / + *

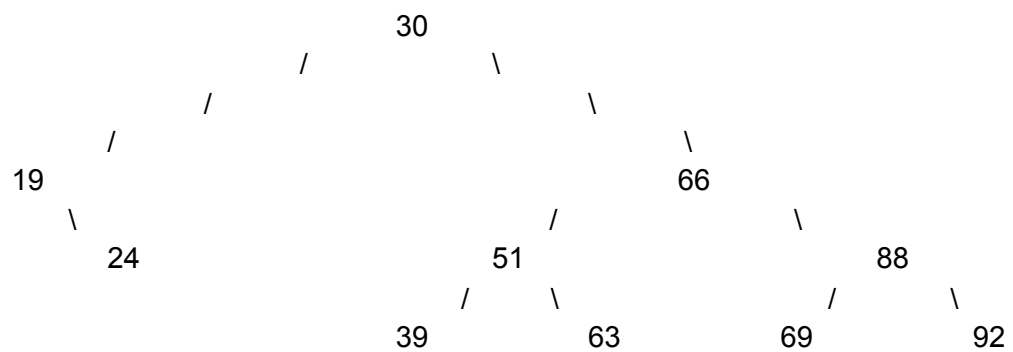
Calculation:

$((13) - (3 \wedge 2)) * ((4) + (21/7)) = (4) * (7) = 28$

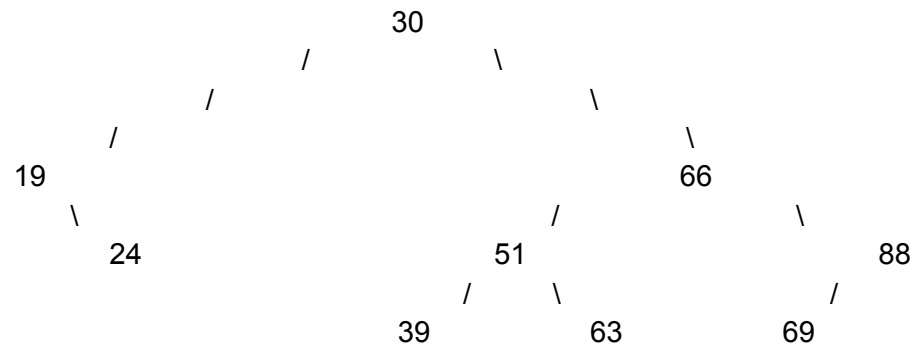
b)



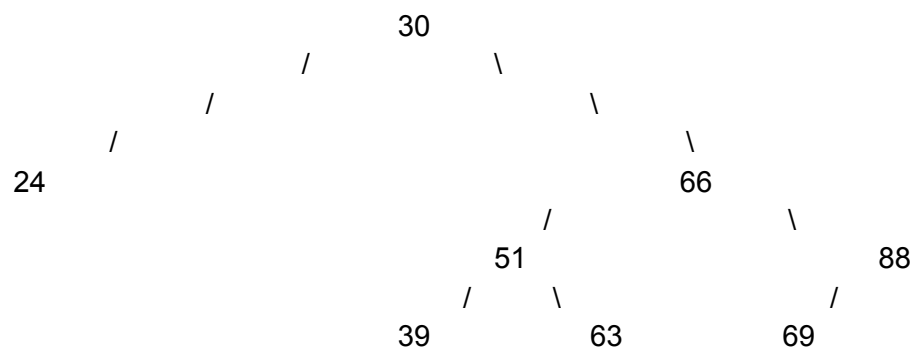
Delete (44)



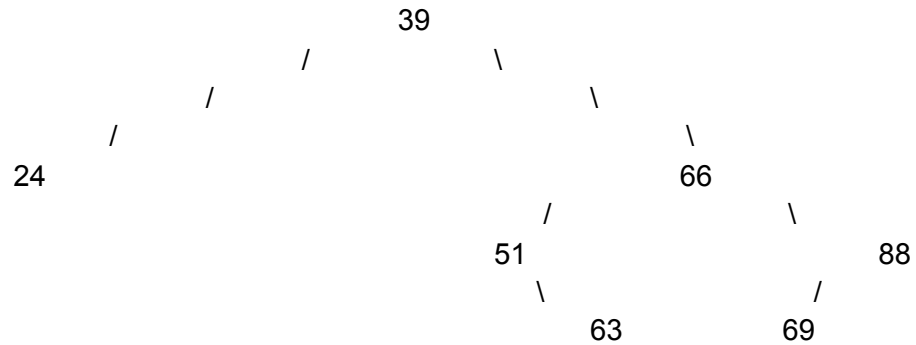
Delete (92)



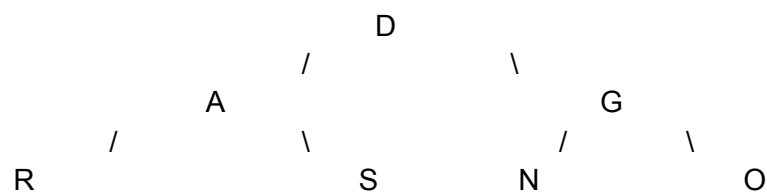
Delete (19)



Delete (30)



c)



Inorder:

R,A,S,D,N,G,O

Q3)

We realize in the figure that there is a significant difference between implementation and theory. For example at 15000 elements we have height of 32 which means we can fit 2^{32}

(= 4294967296) nodes. While in theory we need only a height of 14 2^{14} (=16384). We also see at 7500 nodes for insertion and deletion we have heights of 29 and 28 respectively, which means we can fit 2^{28} and 2^{29} (= 268435456 and 536870912) nodes while we only need a height of 13 for this. If I were to enter the values in sorted way this will provide us with the theoretical values provided in the graph. Since this would mean that the tree is complete so there will be no difference of more than one between the last leaves levels.

FIFF

Tree elements vs height, Theory vs implementation

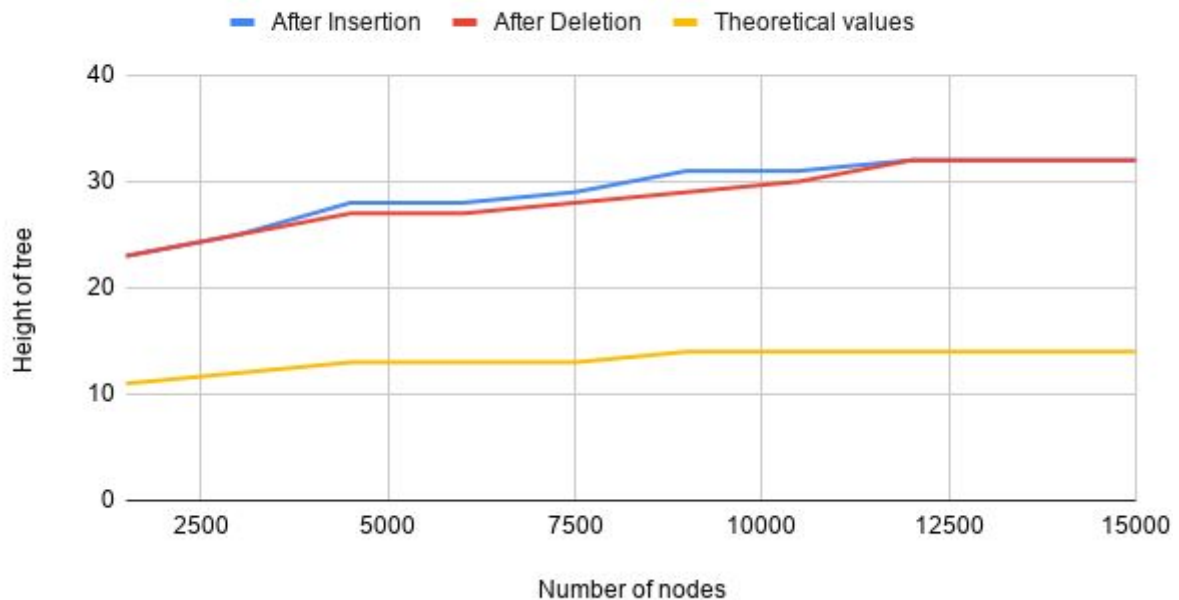


Figure 1: graph of heights of tree, implementation vs theory