CS202
HW2
Section No: 1

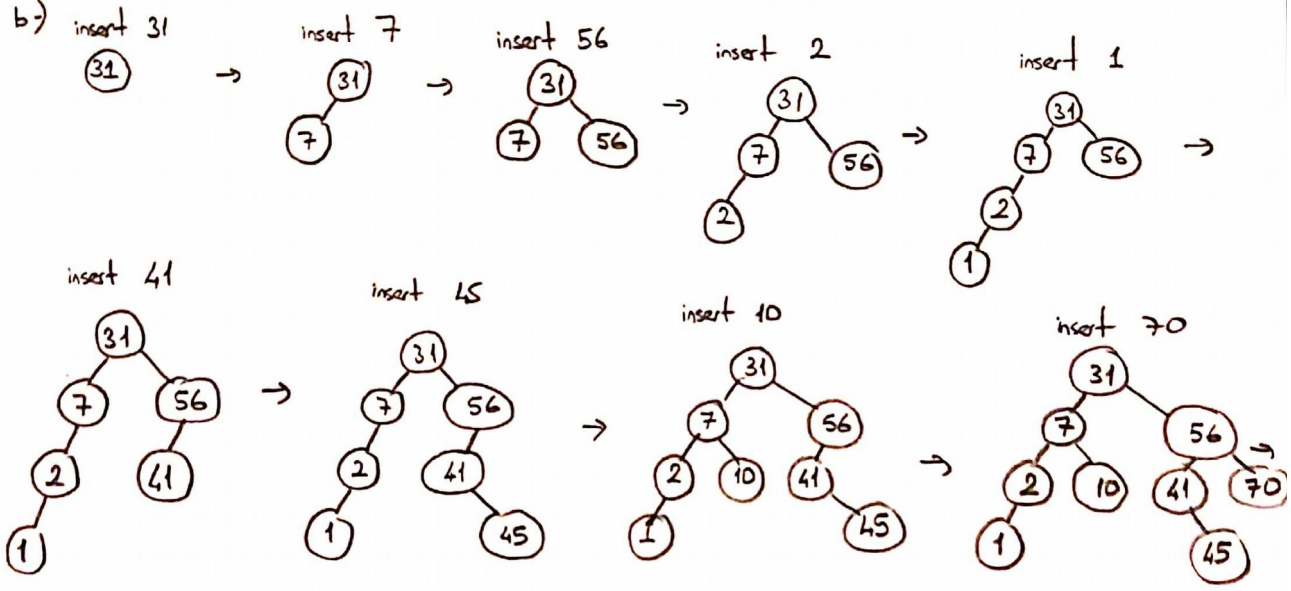
TARIK BUĞRA KARALİ
ID :21703937

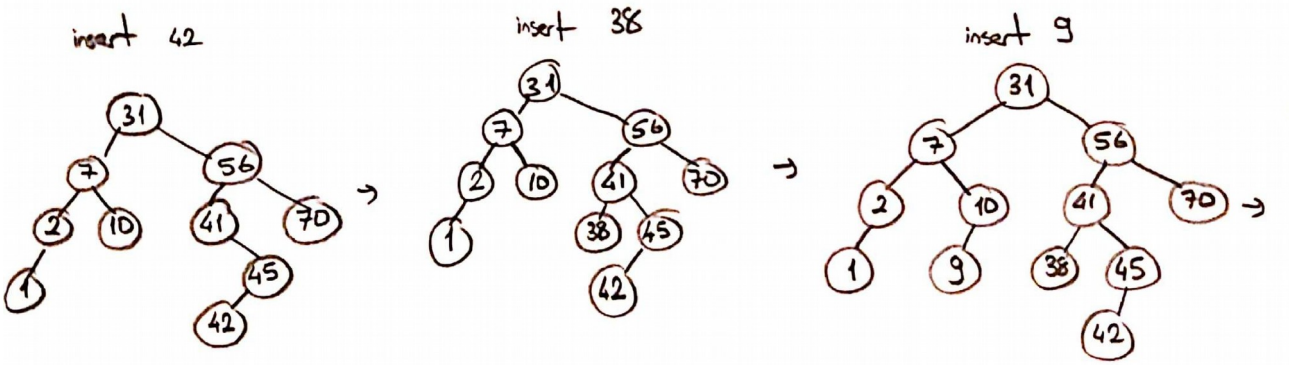
a-) Prefix : /* A + BCD

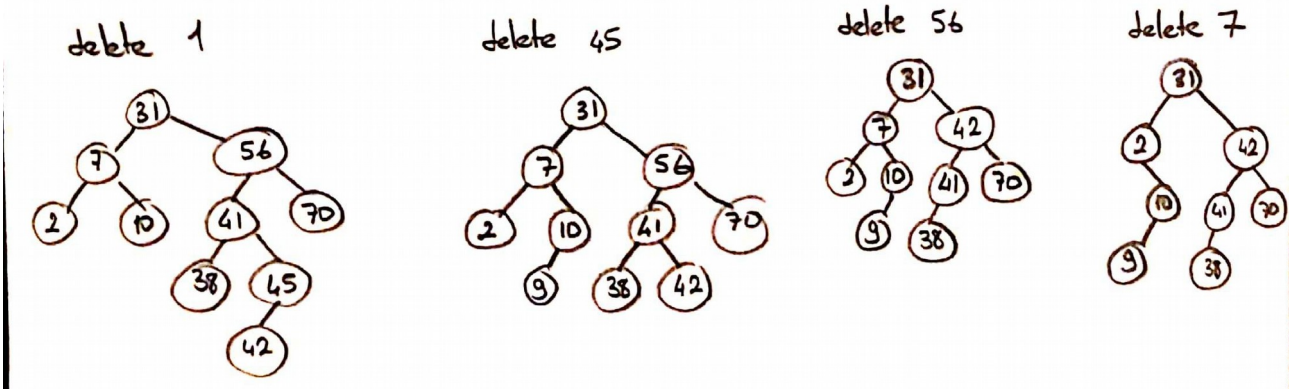
Infix : (A * (B+C)) / D

Postfix : ABC + * D /

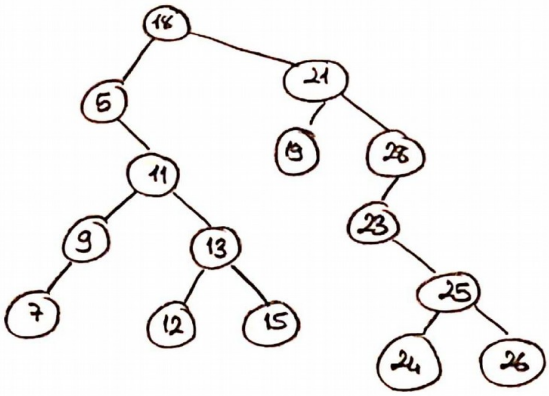
b-)







c-



Post Order Traversal: 7 9 12 15 13 11 5 19 24 26 25 23 28 21 18

LevelorderTraverse: In this method, I use 2 helper methods :

1) printGivenLevel : This method takes the root and height as a parameter and print the items which have same level value recursively.
2) GetNodeheight: This method takes a BinaryNode as a parameter and calculate the level of the node.

In LevelorderTraverse method , firstly we calculate the height of the tree and then using a for loop we print all the level by using printGivenLevel.

Worst Case : O(n^2) . If height is greater than 1 ; we use printGivenLevel ; O(n) + O(n-1) + O(n-2) + .. + O(1) which is O(n^2).

Best Case : O(n) . If height is 1 ; we just use GetNodeheight (O(n)) and cout instruction(O(1)).
O(n) + O(1) = O(n)

Span: I use 1 helper method :rangeDecision

rangeDecision takes as a parameter root and the range limits. It contains three states

1- if root has an int which is greater than lower limit : recursive call for left subtree
2- if root has an int which is in the limit : print value
3- if root has an int which is greater than upper limit : recursive call for right subtree

In the span function we call rangeDecision with parameters root and limits.

Time Complexity: O(n)

Mirror: I use 1 helper method : mirrorHelp

Mirror help takes a BinaryNode as a parameter and then call Mirror for left- BinaryNode(given parameter) , then call Mirror for right-BinaryNode(given parameter) and then change left and right subtrees.

And in the Mirror function Just write the root as a parameter in the MirrorHelp function.

Time Complexity: O(n)