```
/**
* Title : Binary Search Trees
* Author : Berk Yıldız
* ID: 21502040
* Section : 2
* Assignment : 2
* Description : Answers to Question 1 and Question 2
*/
```

1.

a. Preorder Traversal: /-3*58+^417

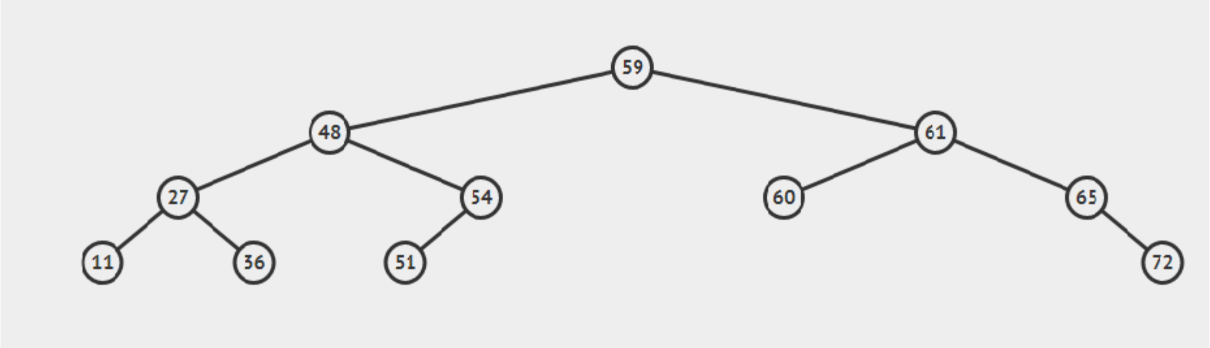
Inorder Traversal: 3-5*8/4^1+7

Postorder Traversal: 358*-41^7+/

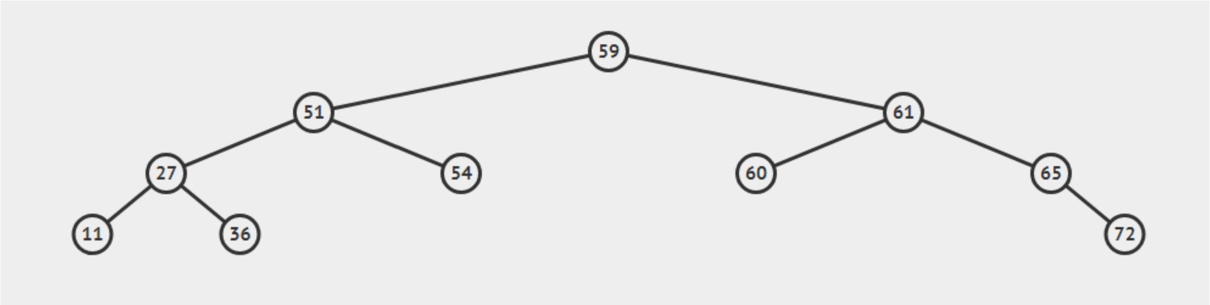
Computation by inorder: (3-(5*8)) / (4^1+7) = (3-40) / (4+7) = -37/11

b.

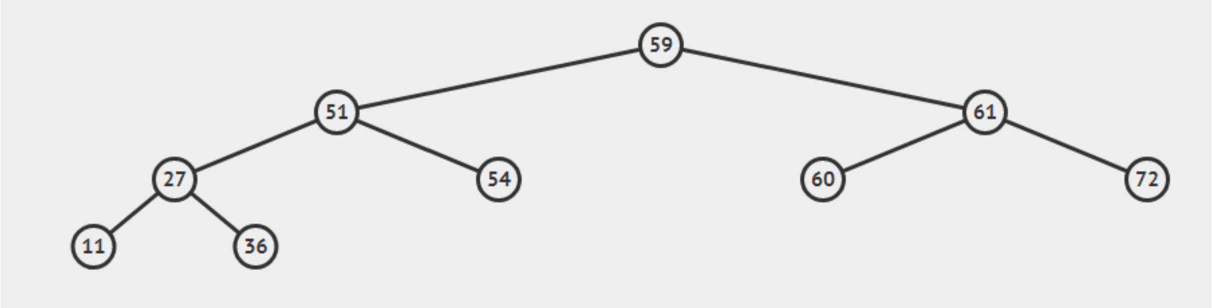
Final tree with all insertions



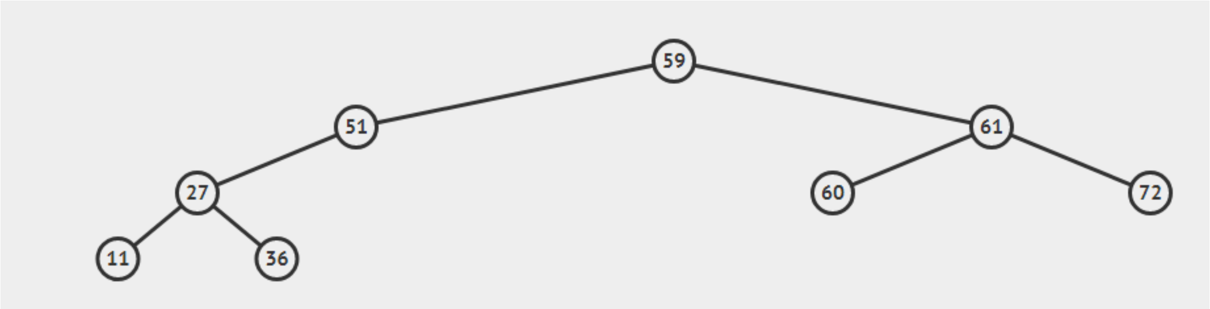
After removing 48



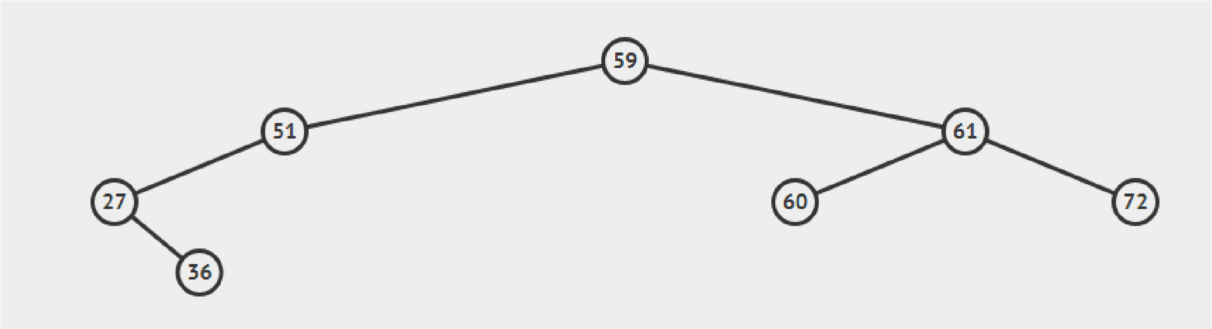
After removing 65



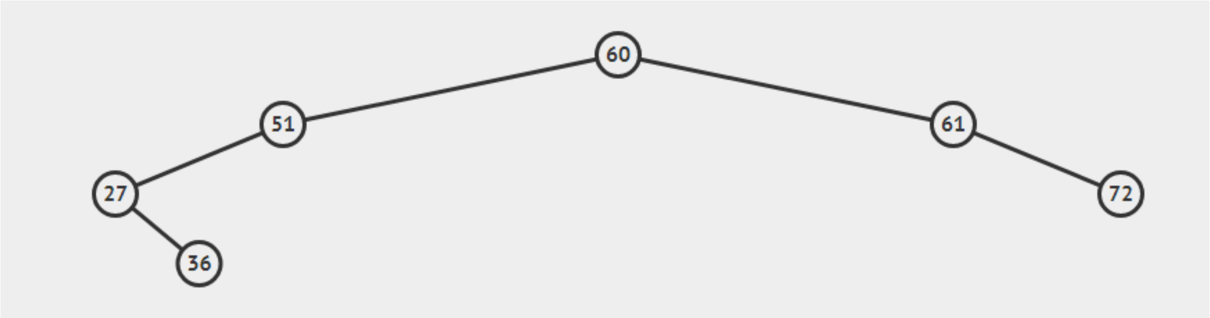
After removing 54



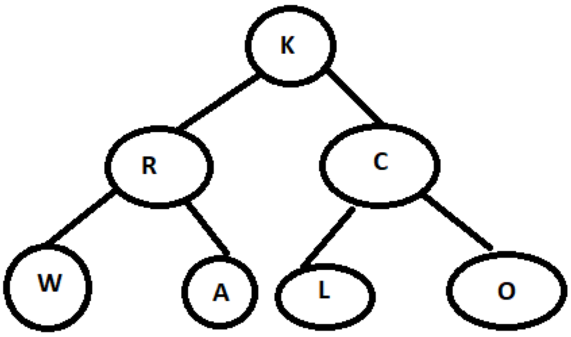
After removing of 11



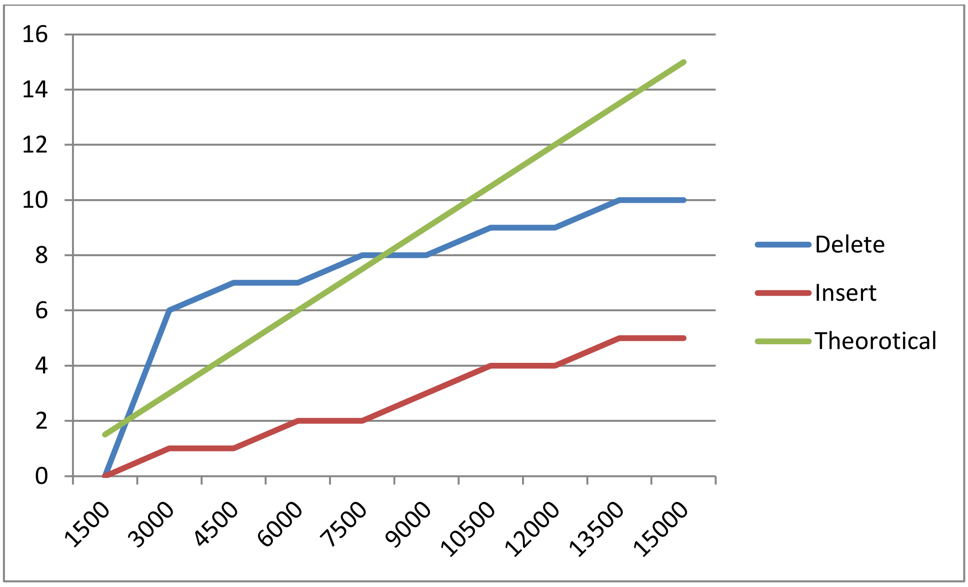
After removing 59



c. Inorder Traversal : W R A K L C O



3)



The graph above gives the result of analysis of insertion and deletion 15000 items to binary search tree. Theorotically inserting n items to binary tree and deleting n items from a binary search tree is O(n). In my results also the graph of deletion and insertion is about the be linear if I make same analysis with larger n values. The differences can be occur from computer the computer because of the different hardware components. The time complexity would be again O(n) if we use sorted values because each time pointer has to reach the end of the tree. But if we design our implementation as inserting all values at begininning and deleting from the beginning the time complexity would be O(1).