Mehmet Onur Uysal

22002609

CS202-Sec1

Homework 2

Question 1

a) Give the prefix, infix, and postfix expressions obtained by preorder, inorder, and postorder traversals, respectively, for the expression tree.

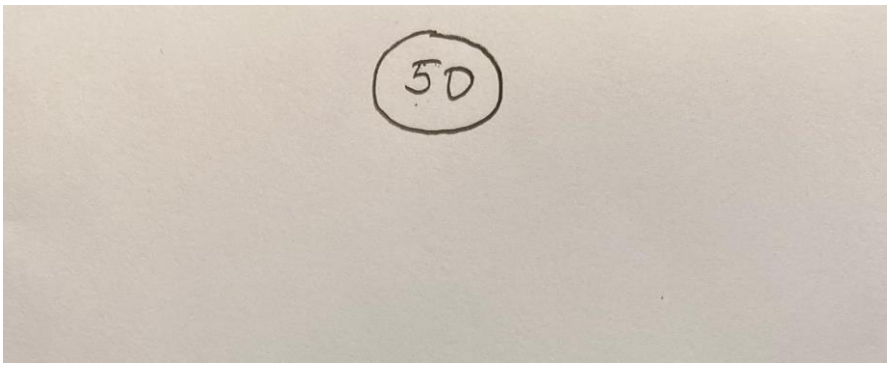
Prefix Expression : X U A B ∩ / C D E

Infix Expression : (A U B) X ((C / D) ∩ E)

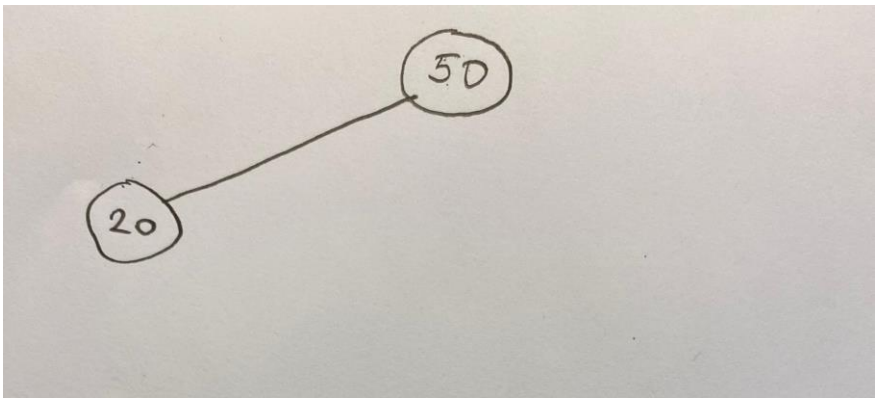
Postfix Expression : A B U C D / E ∩ X

b) Insert 50, 20, 80, 10, 65, 75, 45, 90, 70, 60, 30, 40, 63 to an empty Binary Search Tree, and then delete 10, 75, 80, 20, 50 in the given order. Show the evolution of the BST after each insertion and deletion operation.

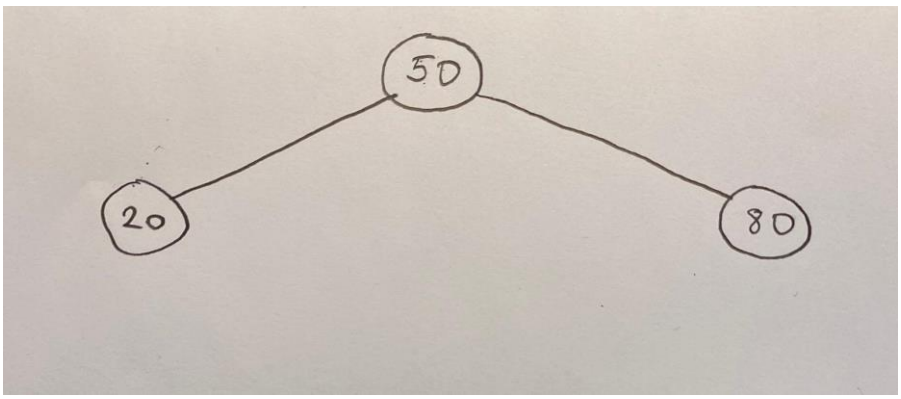
After inserting 50 to an empty binary tree



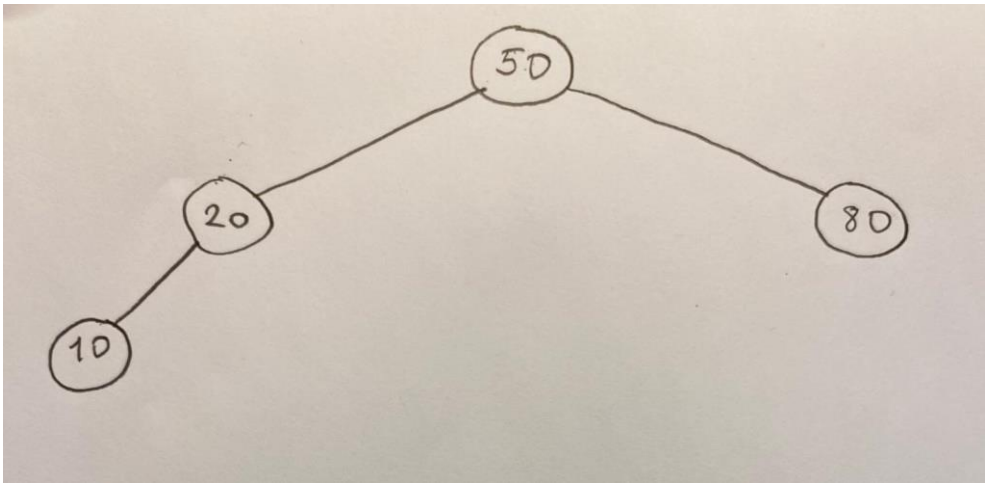
After inserting 20



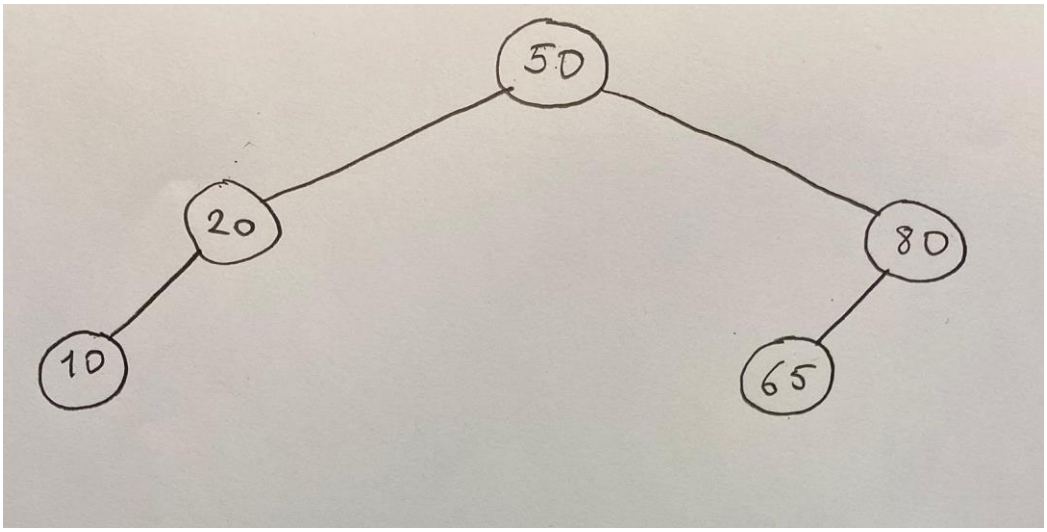
After inserting 80



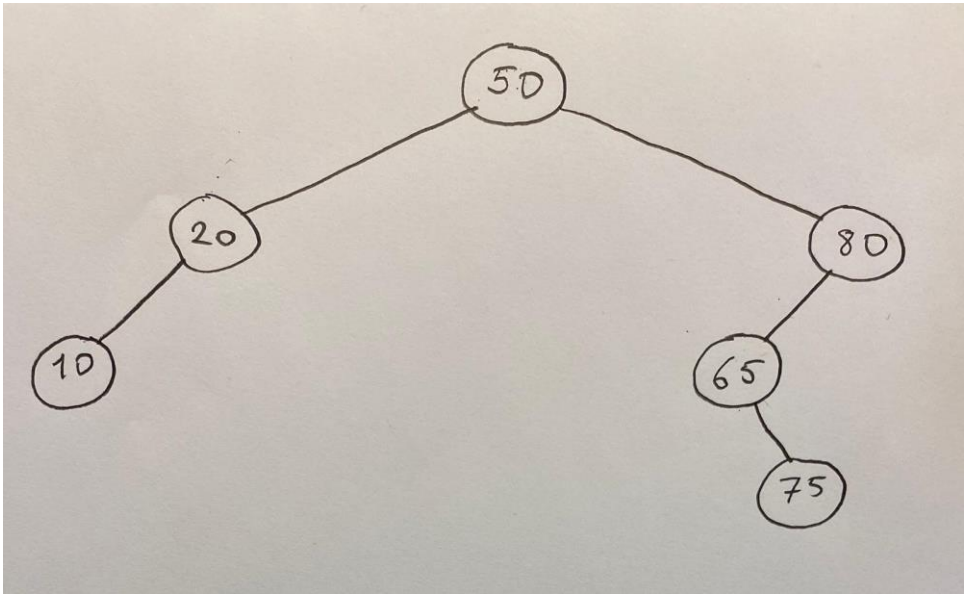
After inserting 10



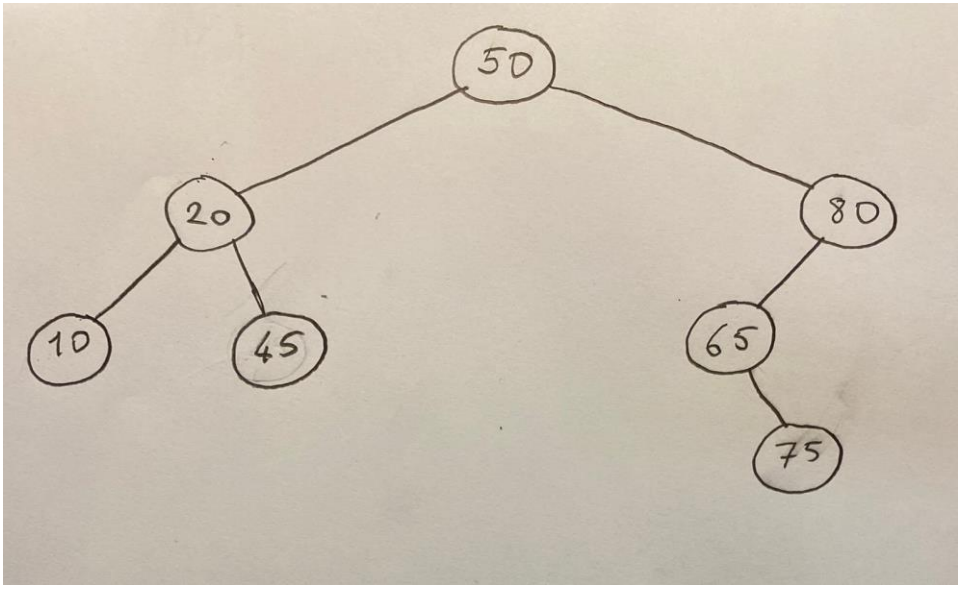
After inserting 65



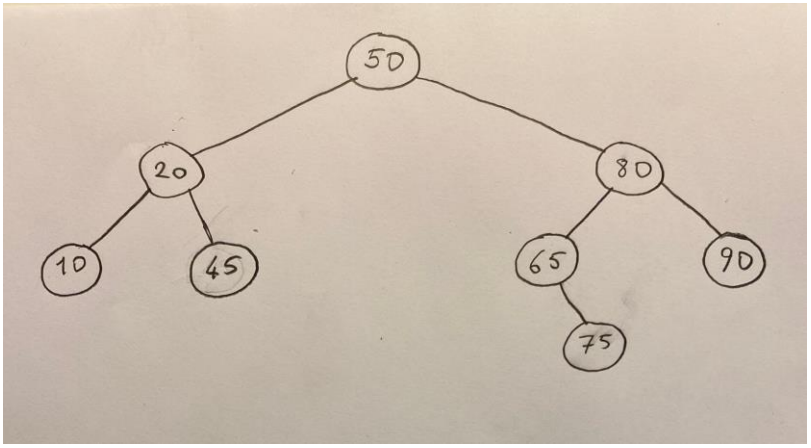
After inserting 75



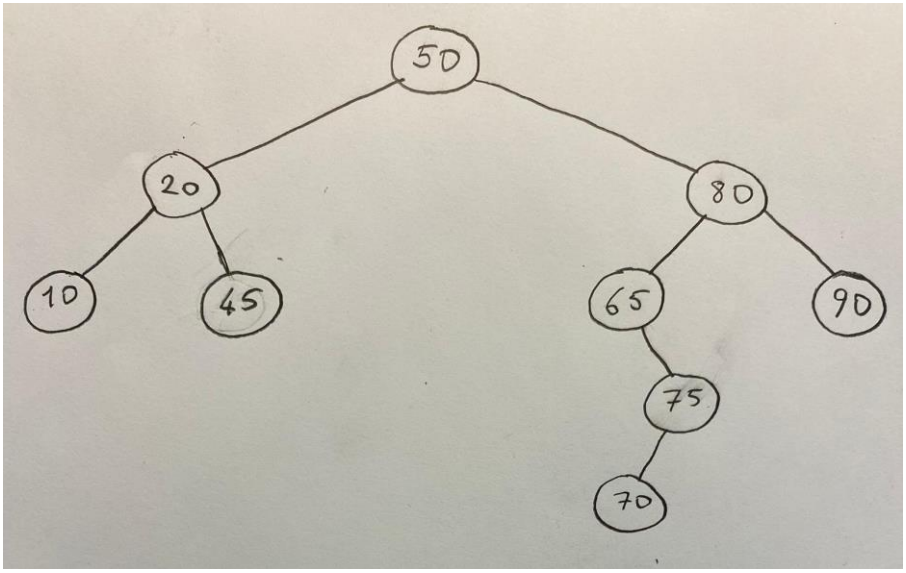
After inserting 45



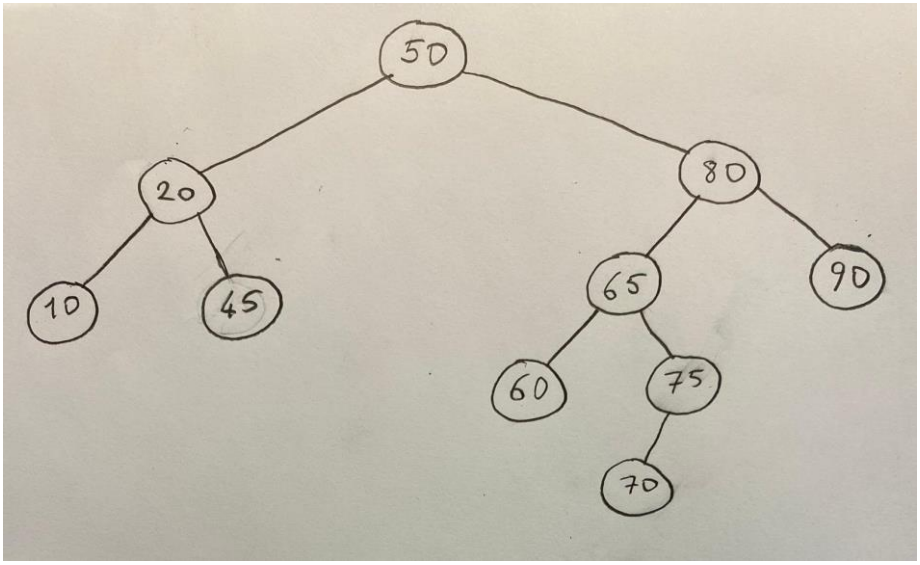
After inserting 90



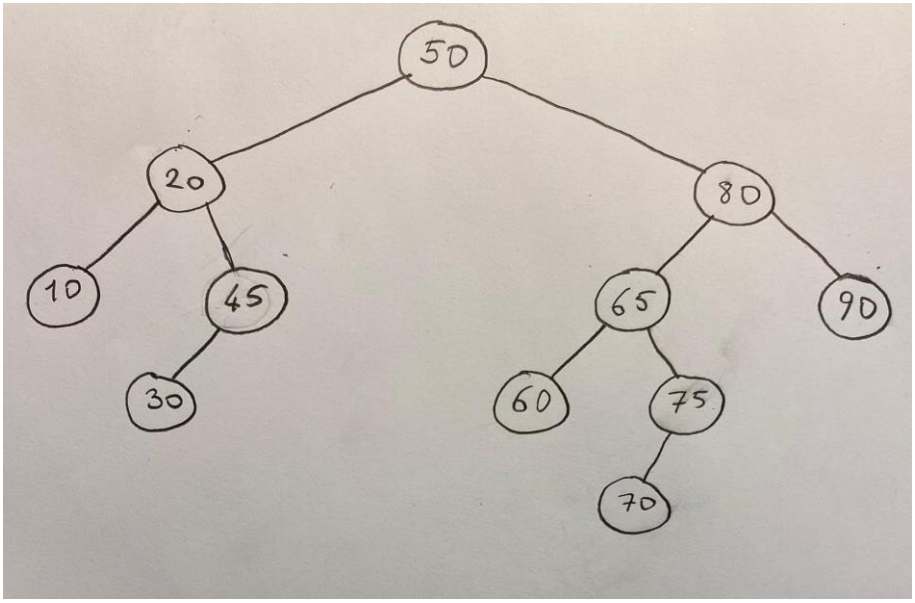
After inserting 70



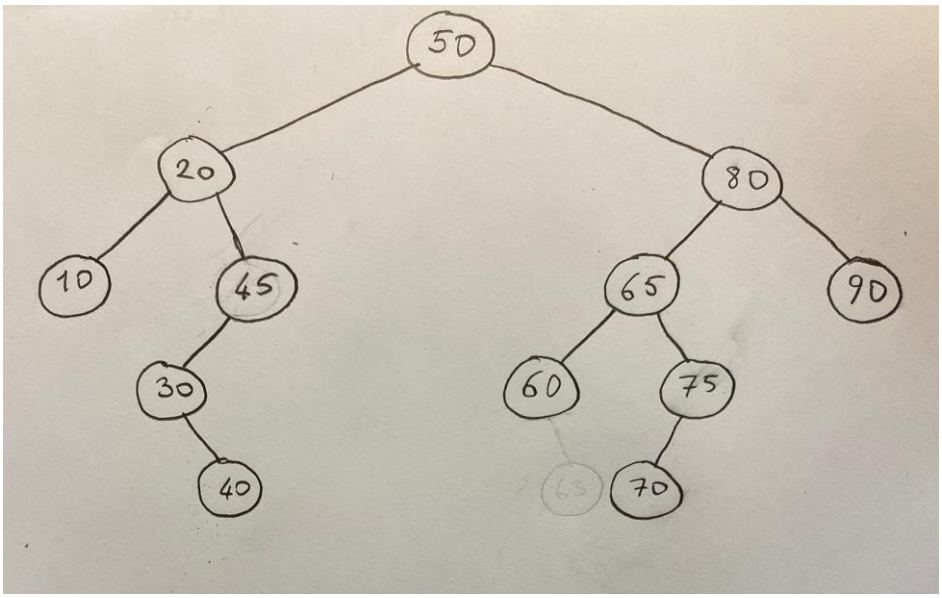
After inserting 60



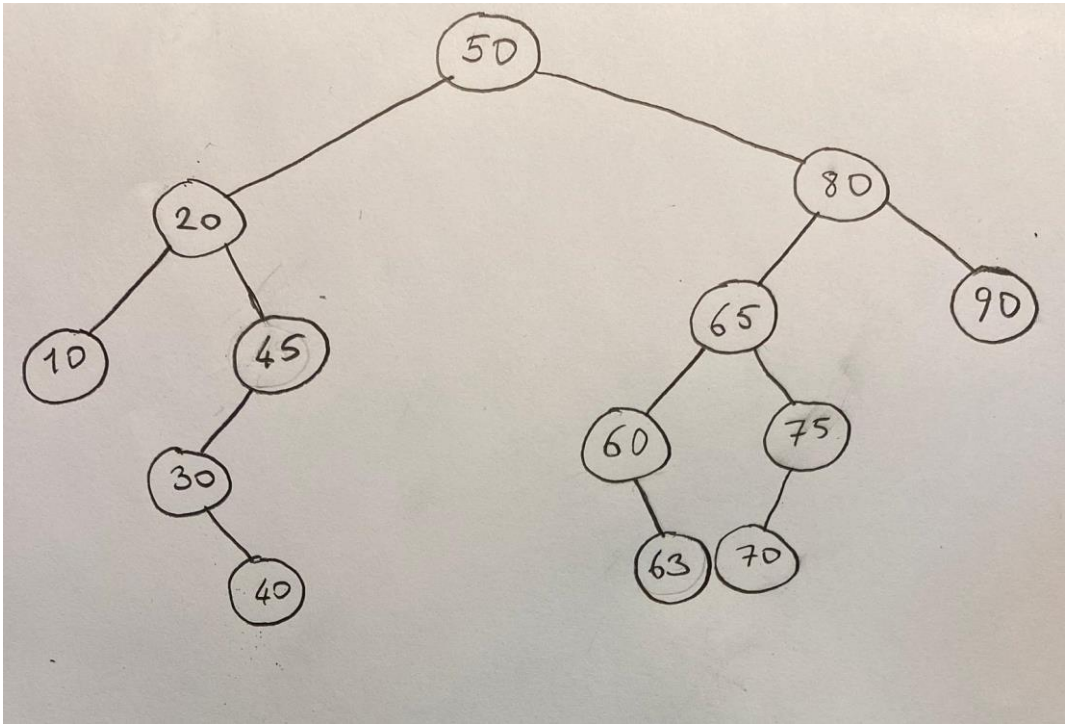
After inserting 30



After inserting 40



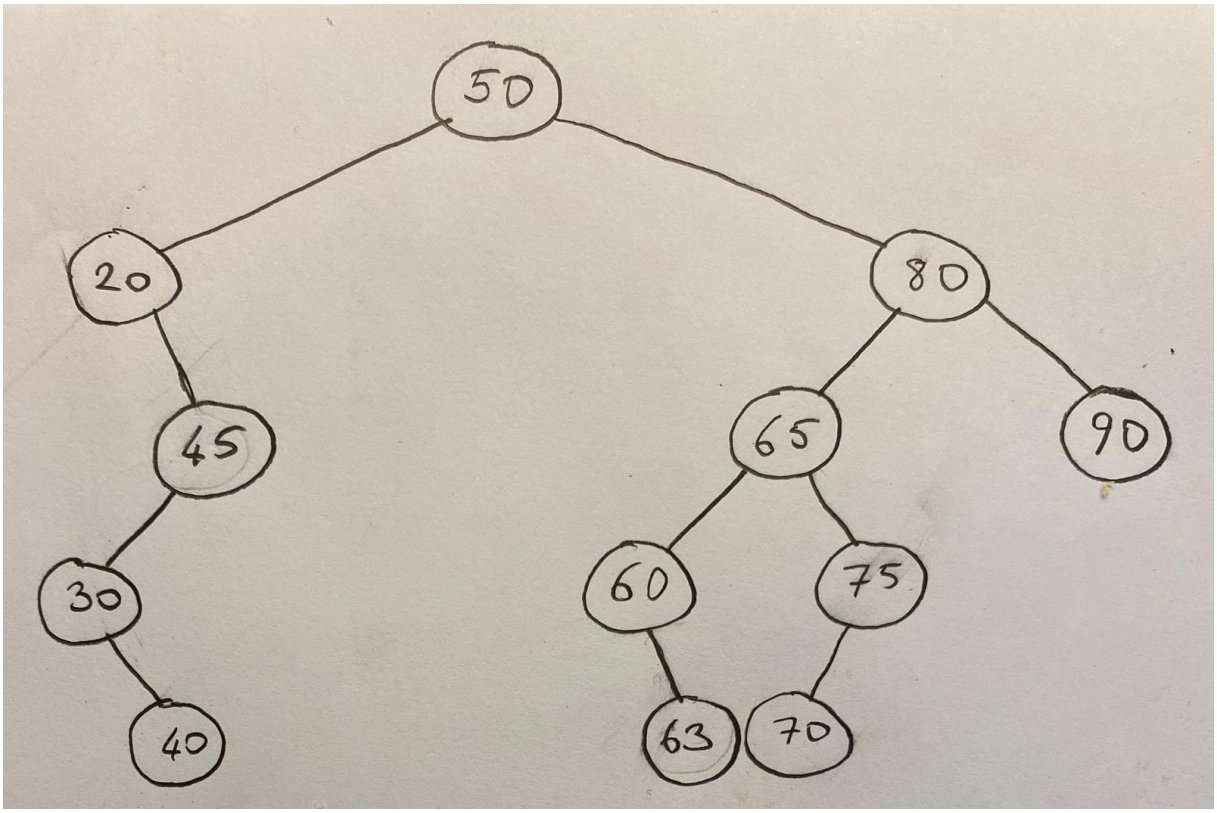
After inserting 63



Deletion

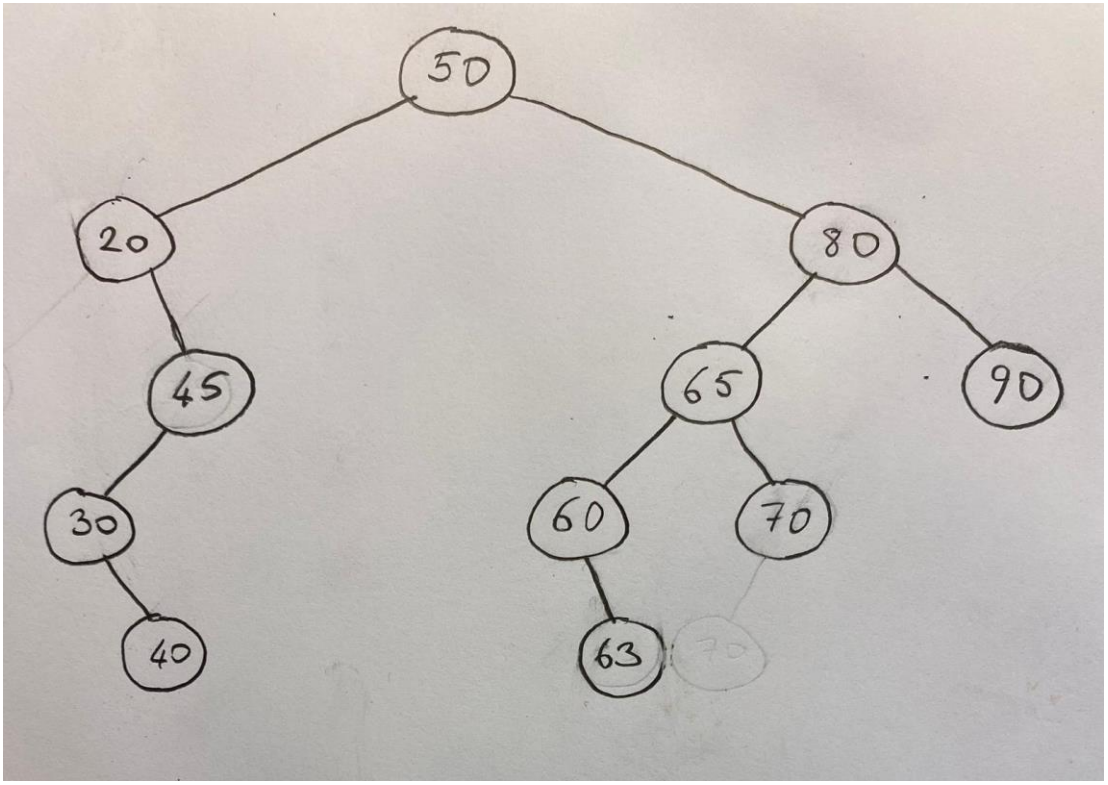
10 is on a leaf node. So, we can simply delete it and set it's parent left node to null.

After deleting 10



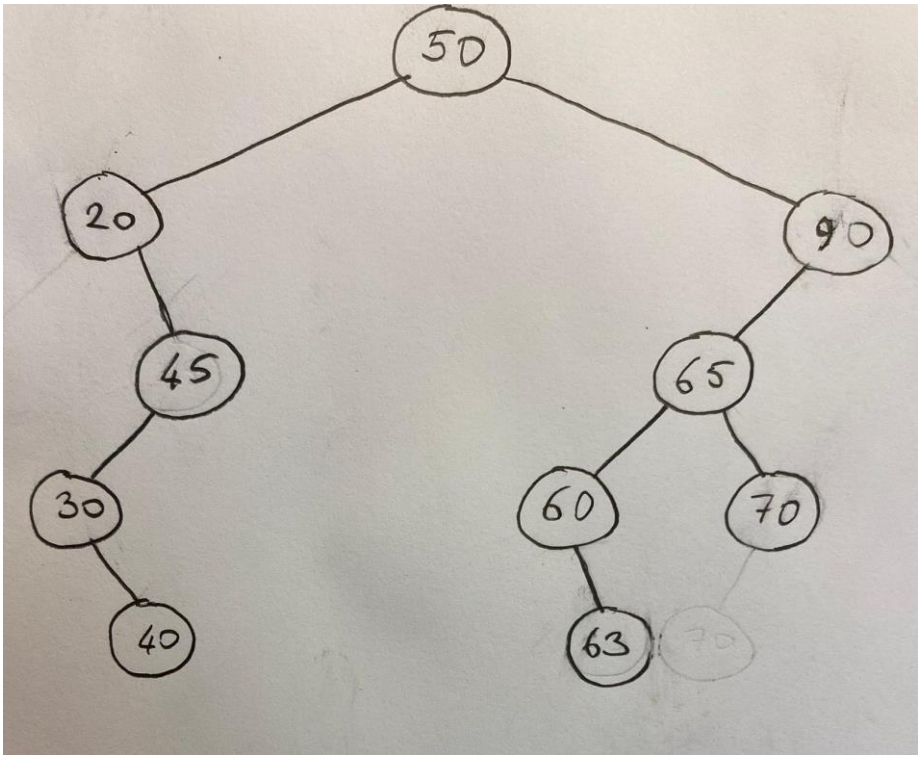
75 has only one child so we can replace 75 with it's child and then delete

After deleting 75



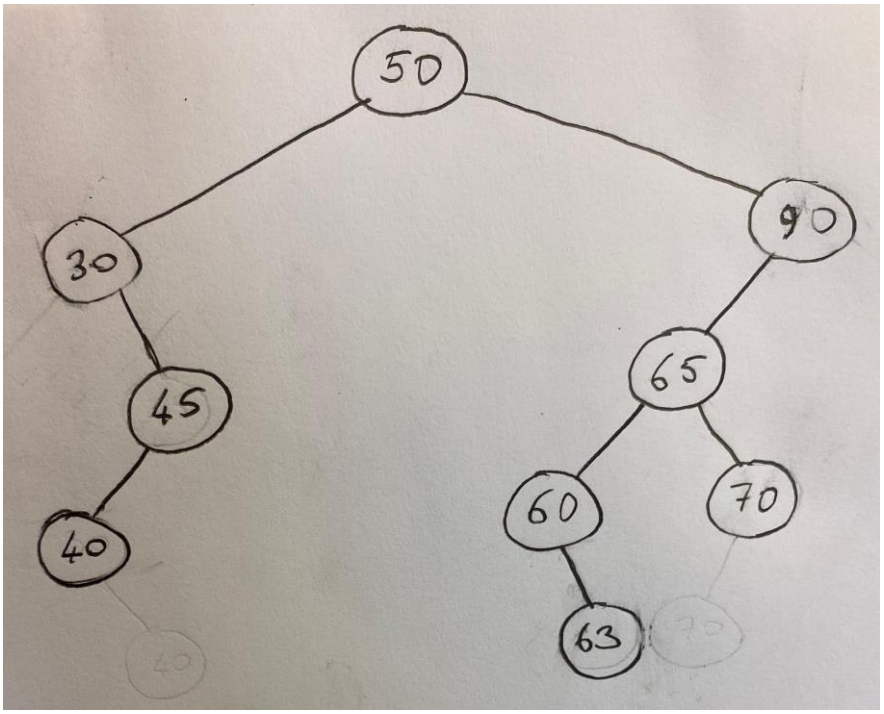
80 has two children so, we have to find the minimum number that is greater than 80 (which is currently 90), swap them and then delete 80

After deleting 80

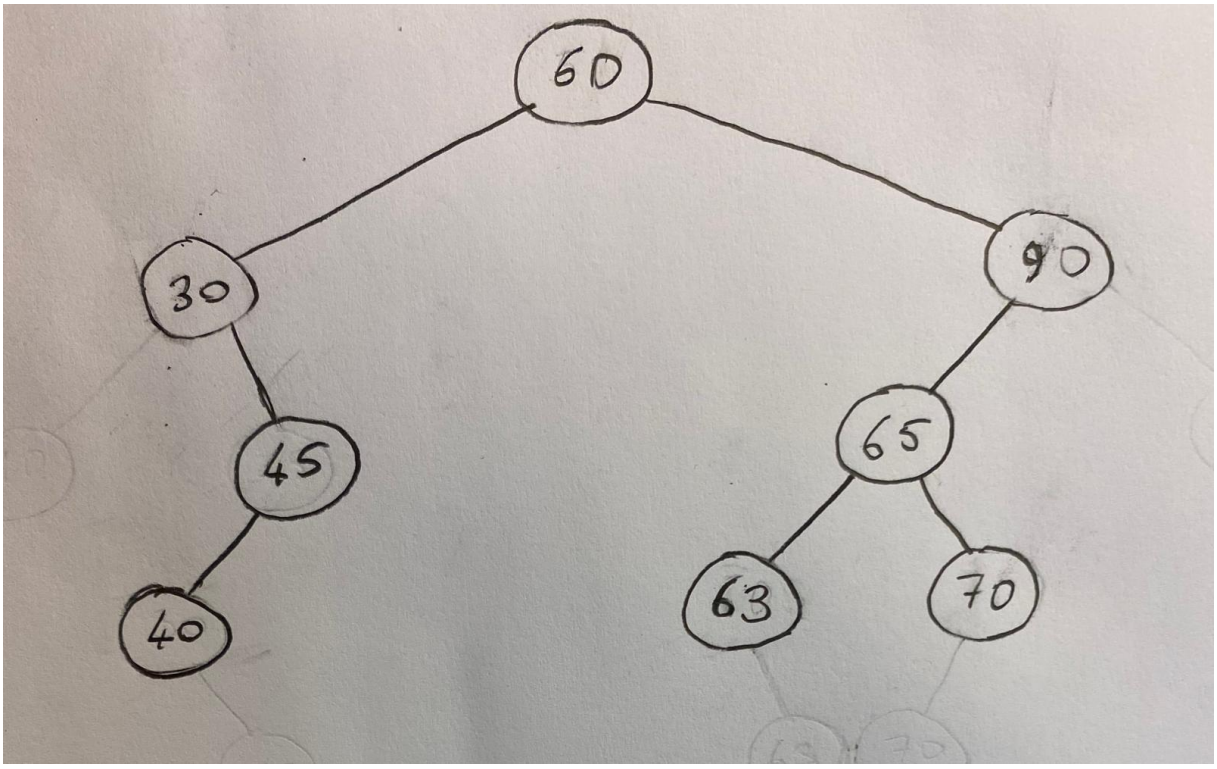


20 has one child since 10 was previously deleted. So, we can delete it and put it's child there.

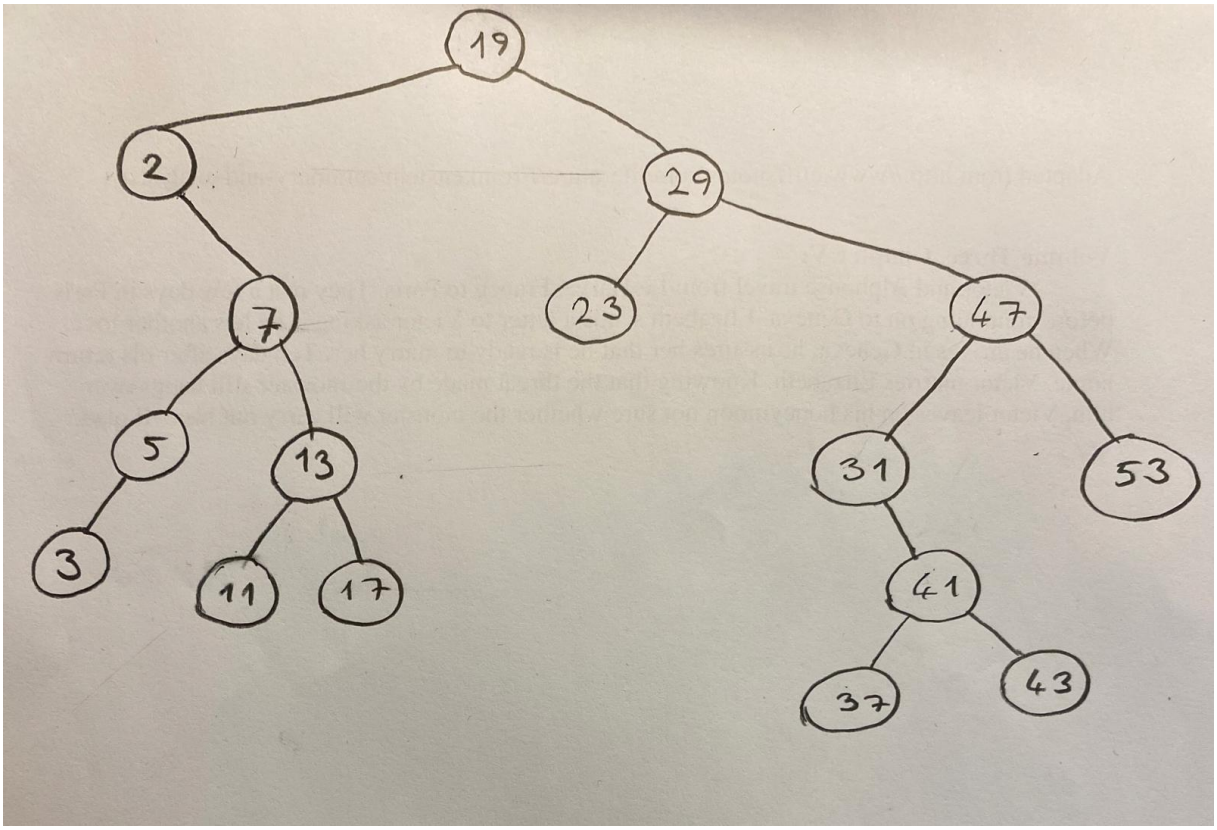
After deleting 20



50 is the root and has two children. We have to find the minimum number that is greater than 50, which is 60, swap them and then delete. 60 also has a child so we will replace it with its child and then delete.



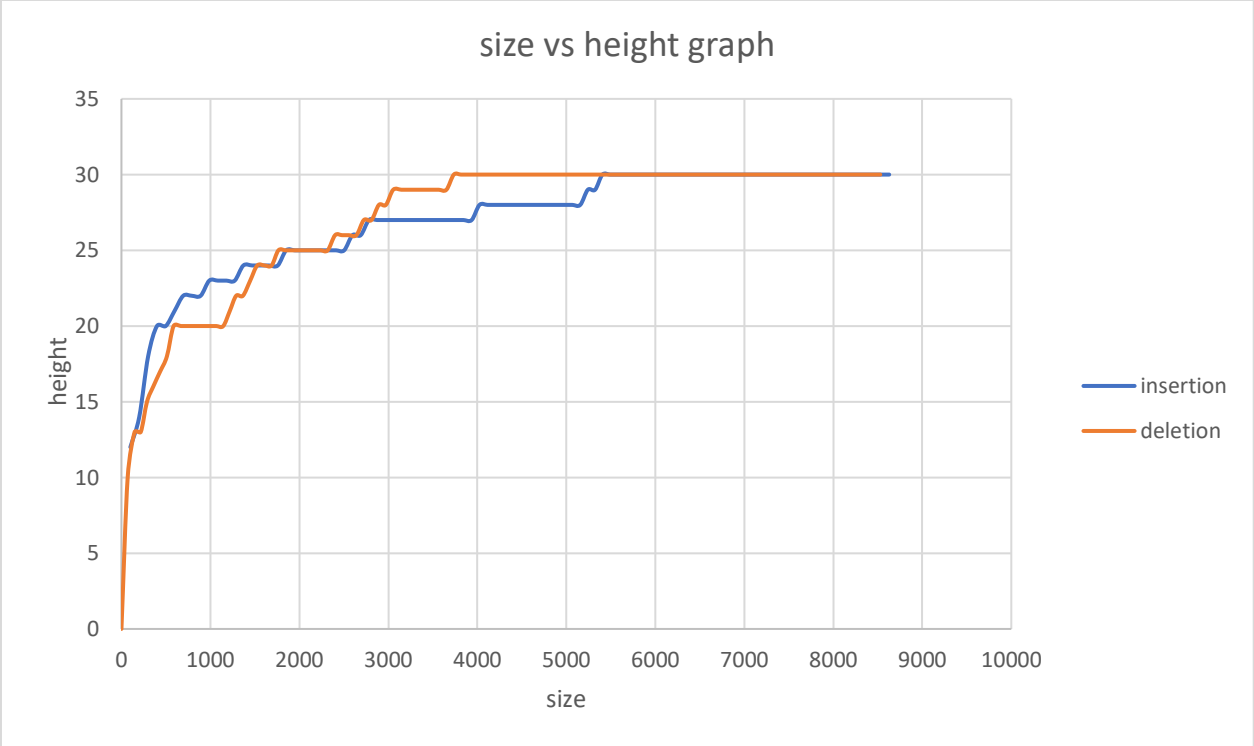
C) A binary search tree has a postorder traversal of 3, 5, 11, 17, 13, 7, 2, 23, 37, 43, 41, 31, 53, 47, 29, 19. Give the corresponding binary search tree. What is its preorder traversal?



Binary tree of the given postorder traversal

Preorder traversal : 19,2,7,5,3,13,11,17,29,23,47,31,41,37,43,53

Question 3



The graph above shows the change in the height of the binary search tree as a function of the size of the tree. Two lines are for insertion and deletion operations respectively. The tree was created using 10000 random integers in the range 0-10^{12}. Due to some duplications, tree had the size 8628 after insertion operations.

It is clearly seen from the graph that for both insertion and deletion height changes logarithmically with respect to the size. If we think very simple, it makes sense that the growth of height decreases because at each height, the tree can take two times the integer values of the previous height. So, two times the integers are required to increment the height compared to the previous step.

Efficiency of most binary tree operations depend on the tree height and the minimum height of a binary tree with n nodes is $\log_2(n+1)$. Insertion and deletions are two of the operations that depends on the height of the tree for time complexity. For the binary search tree, insertion and deletion operations have worst case time complexity O(N) and average case O(log(N)). Efficiency of these operations increases when the tree is more balanced. Since we added random integers and did the experiment with a quite large tree (with 8628 nodes), it is expected that the graph would show logarithmic behaviour. We can confirm this with the graph above and say that this was a successful experiment.