Mehmet Bora Kurucu

CS202-Section 1-HW2

21703404

Question 1)

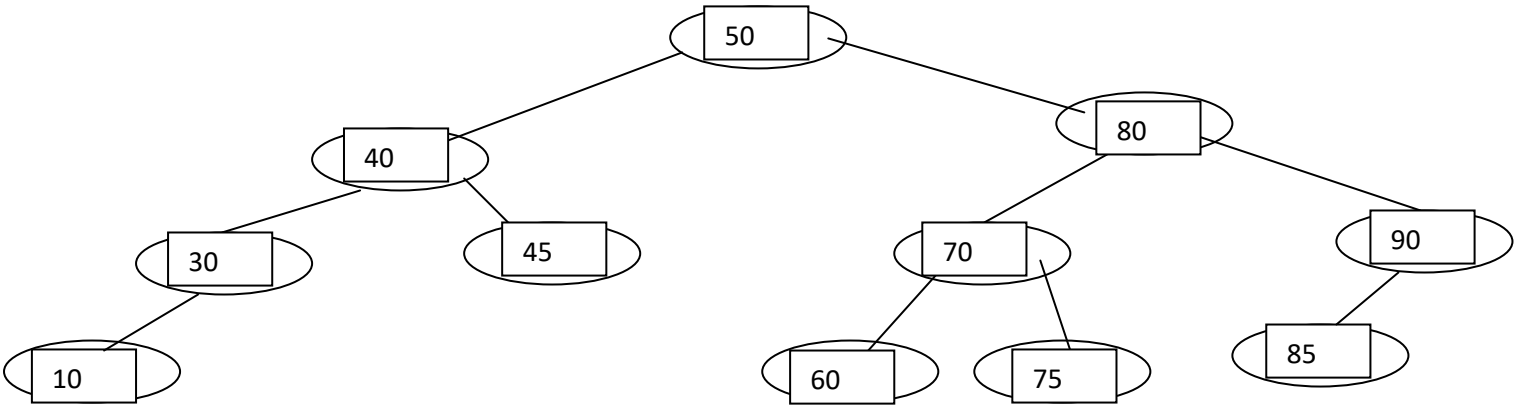
a)

Preorder: 1,2,4,7,8,5,9,10,12,13,3,6,11

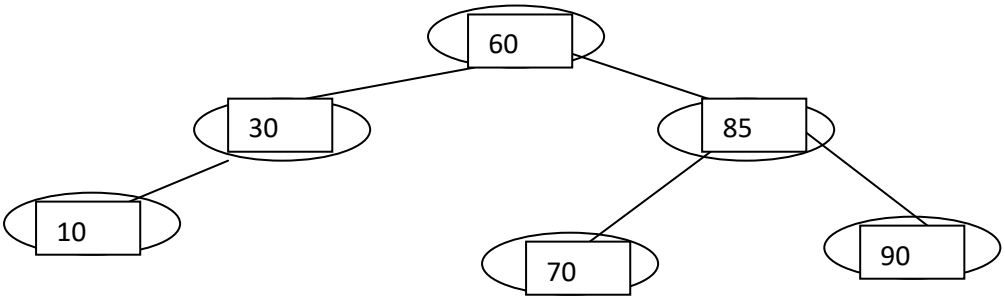
inorder: 1,2,4,7,8,9,5,12,10,13,3,6,11

Postorder: 7,8,4,9,12,13,10,5,2,11,6,3,1

b) Insertion:



Deletion:



Question 3)

Worst case of Binary Search Tree's ocur when Binary Search Tree's are linearly ordered. Consider inserting 10,11,12,13 to a Binary Search Tree, respectively. When this happens, the essential advantage of binary search trees,division,halving the input size each time and operating on the halved part is removed. The tree behaves like a linked list, as there is no clever solution, you can't just divide it and work on the divided half,since there is no half. Time complexity of the addNgram in big theta notation is Θ(n). Imagine BST's nodes are 10,11,12,13, and one wants to insert 14. All of the list needs to be traversed one by one without any division,without halving the input size, for this operation Θ(n). Than, node is inserted with a simple connection. Time complexity of printNgramFrequencies is also Θ(n). Since, one has to traverse all of the list one by one ,without any halving when the BST is lineraly sturctured.

To be more technical:

$T(n) = T(n/2) + 1 = \Theta(\log n)$ tranforms into $T(n) = T(0) + O(n) = \Theta(n)$ in worst case.