Oğuz Kaan İmamoğlu

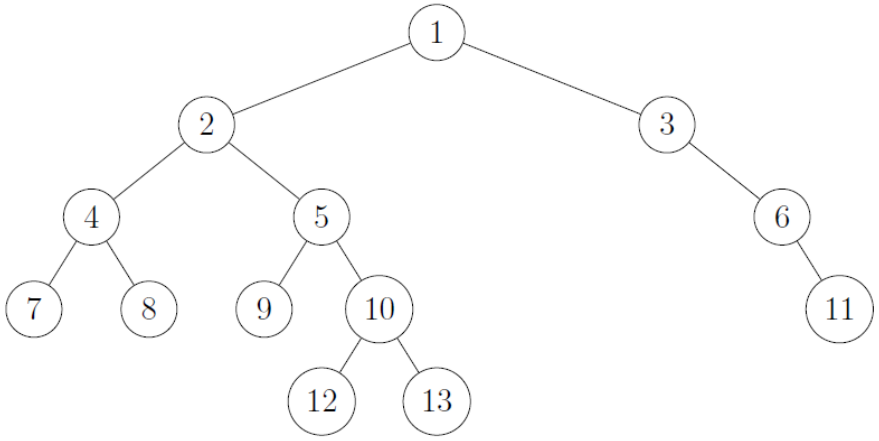
21702233

CS202 – 01

Homework 2

Question 1:

a) What are the preorder, inorder, and postorder traversals of the binary tree below:



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

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

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

b) Insert 50, 40, 80, 30, 45, 70, 90, 10, 60, 75, 85 to an empty Binary Search Tree. Show only the final tree after all insertions. Then delete 45, 40, 80, 75, 50 in given order. Show only the final tree after all deletion operations. Use the exact algorithms shown in lectures.

Figure 1 shows the bst after all insertions.

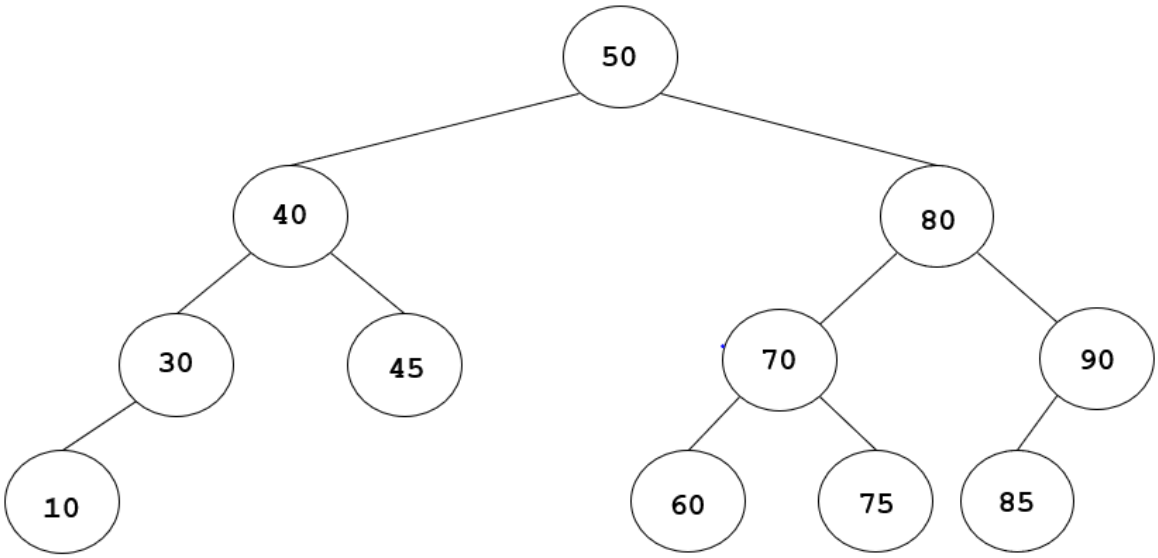


Figure 1

Figure 2 shows the bst after all deletions.

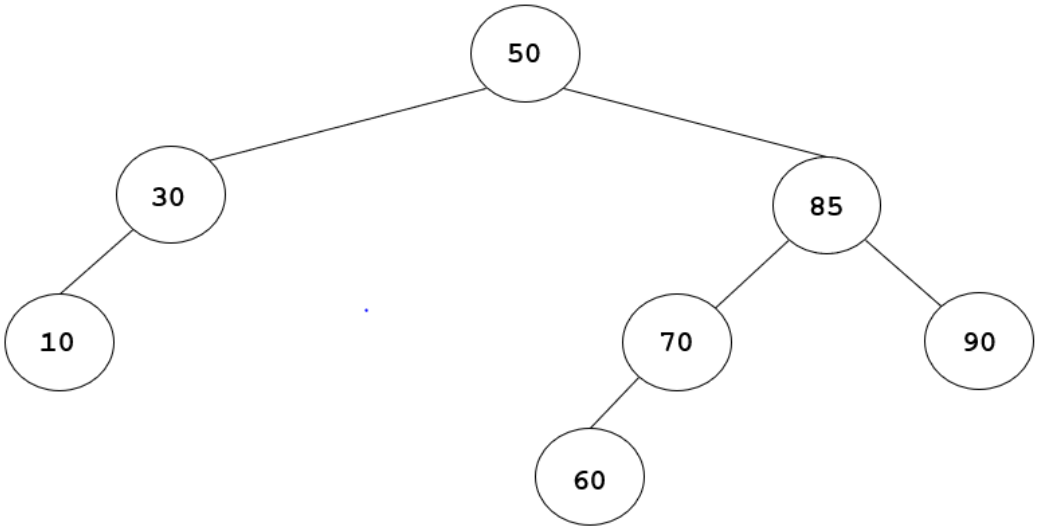


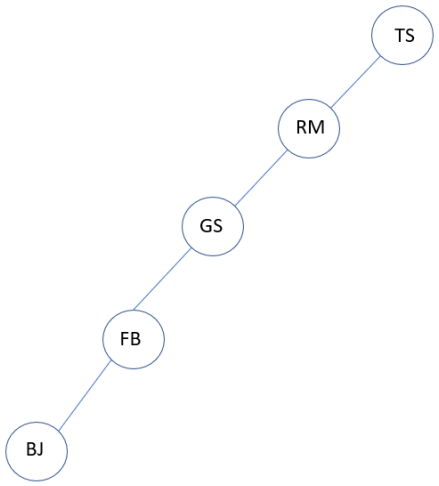
Figure 2

Question 3:

Let us begin with printNgramFrequencies function. In order to calculate the how many repetition does a item, we must visit each repeated item. Since we must calcute frequencies of each significant item in the tree, this means we must visit every item in the tree. It means:

Worst case running time of printNgramFrequencies is Θ(n).

For addNgram the initial organization of the tree is crucial. Running-time complexity depends on that. Since we are calculating the worst-case running time we must consider the worst organization, which is an unbalanced, linear-like tree. Like that:



In this example there are no right sub-tree and it looks like an array. It means that we have to traverse through all the tree in order to insert a new item. For example we will insert AA. We must iterate until we reach BJ. So we can say that:

Worst case running time of addNgram is Θ(n).