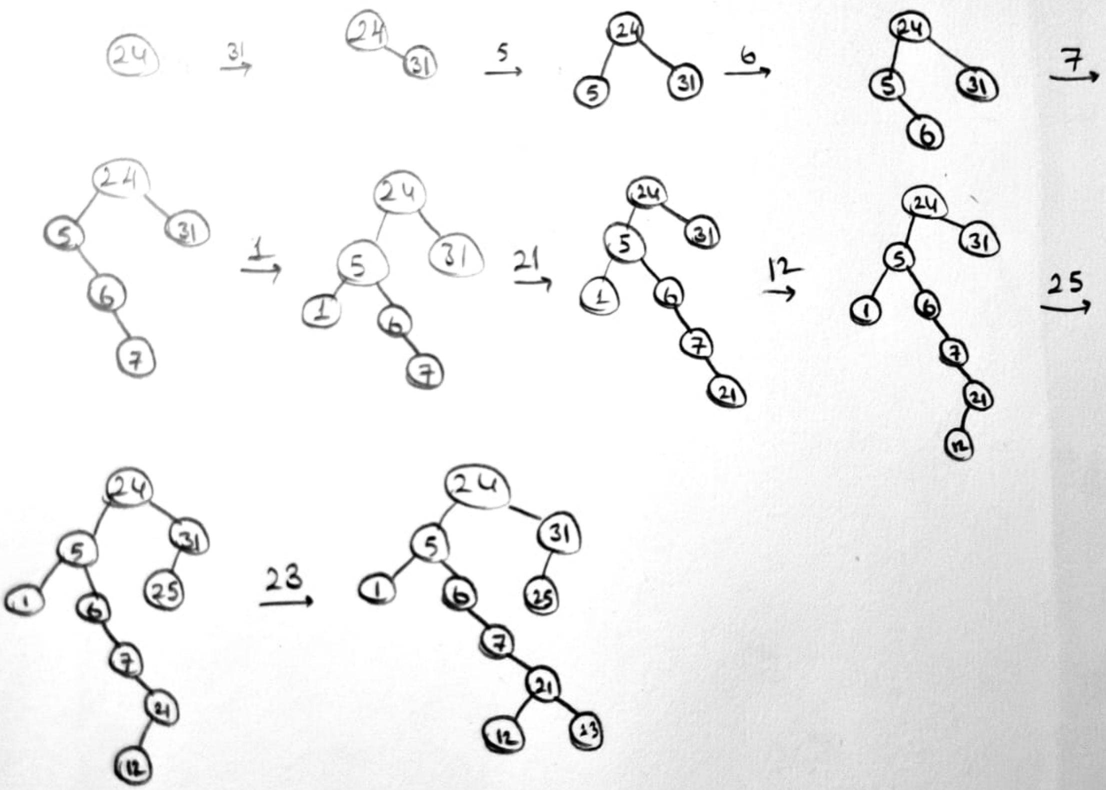
Aral Müftüoğlu
22201566
Question 1)

CS-202 Homework-1
Aral Müftüoğlu
22201566
Section 3

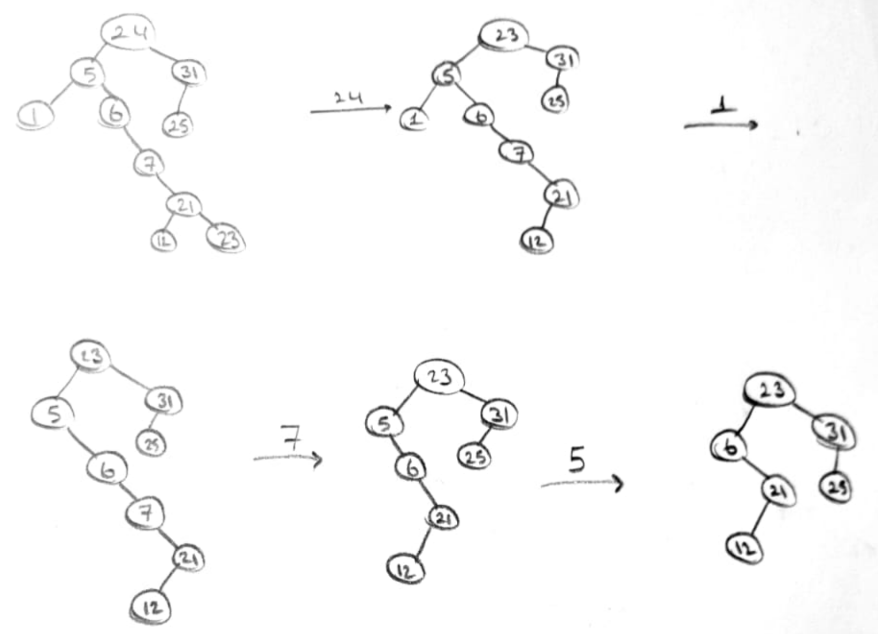
Question 1

A) Insert 24, 31, 5, 6, 7, 1, 21, 12, 25, 23

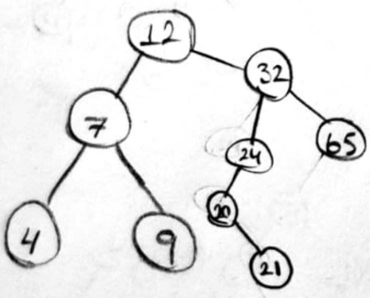


B) preorder traversal → 24, 5, 1, 6, 7, 21, 12, 23, 31, 25
inorder traversal → 1, 5, 6, 7, 12, 21, 23, 24, 25, 31
postorder traversal → 1, 12, 23, 21, 7, 6, 5, 25, 31, 24

C) Delete 24, 1, 7 and 5



D) Postorder sequence ⟶ 4, 9, 7, 21, 20, 24, 65, 32, 12



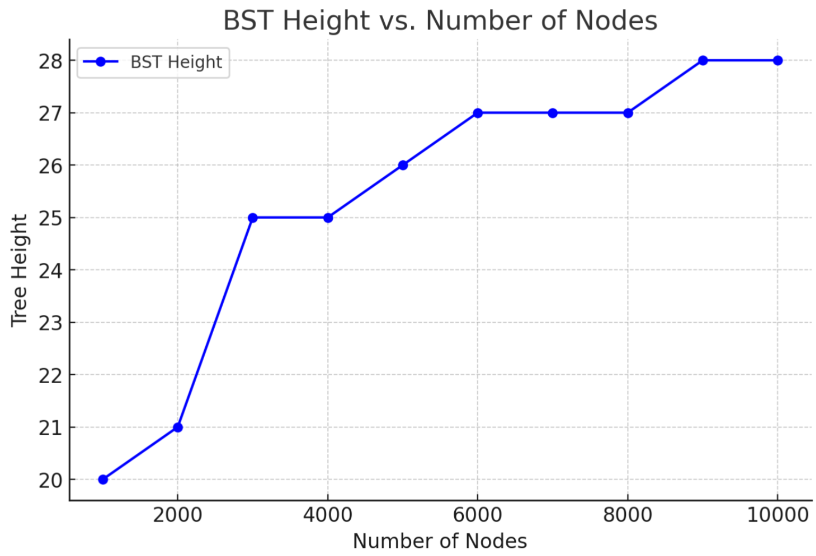
} BST for the elements.

Preorder sequence → 12, 7, 4, 9, 32, 24, 20, 21, 65

6th key in the preorder traversal sequence → 24

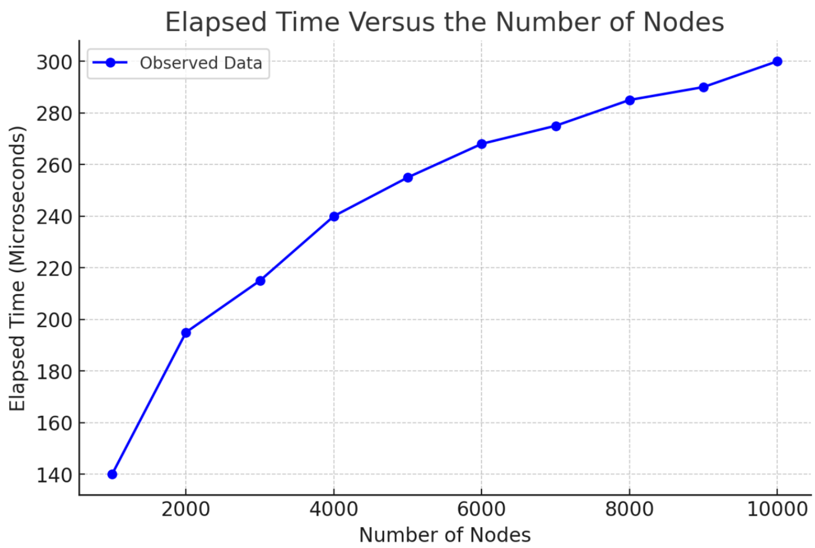
Question 3)

A)



It is not a linear function or logarithmic function. The tree's height graph represents the structural change in the BST during the insertion methods. Starting at a random point and then generally increasing, the height reflects the tree's growing complexity. Unlike a perfectly balanced tree where height would grow very systematically, a standard BST with random insertions will show more randomly altering or unpredictable height progression. As more elements are added the height increase but not in a strictly linear or perfectly logarithmic manner. This result is normal for unbalanced BSTs, where the tree's shape stems from random numbers.

B)



This function looks like a basic log(n) function as expected from implementation. As you add more elements to the tree, the time required to insert each new element increases very slowly, almost logarithmically. This means that even as you approach 10,000 elements, the insertion time remains relatively constant and predictable. If already sorted integers are inserted (either ascending or descending order) into a BST, the time complexity for insertions becomes O(n) instead of the typical O(log n). This means the insertion time grows linearly with the number of elements, which is dramatically different from this one. Each new element will be inserted as a leaf at the end of a linear chain, effectively turning the BST into a linked list-like structure.