CS 202

Homework 2

Sec 03

Kutay Tire

22001787

27.03.2022

TA: Navid Ghamari

Instructor: Ertuğrul Kartal Tabak

Kutay Tire 22001787

Question 1

a.) preorder → prefix expression: $X \cup A B \cap \setminus C D E$

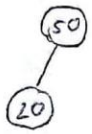
inorder → infix expression: $A \cup B \times C \setminus D \cap E$

postorder → postfix expression: $A B \cup C D \setminus E \cap X$

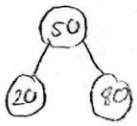
b.)

(1) Insert 50

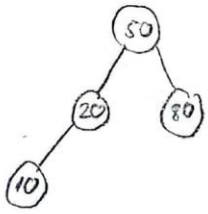
(2) Insert 20



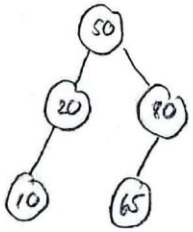
(3) Insert 80



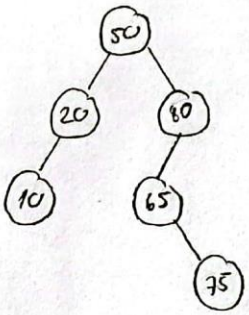
(4) Insert 10



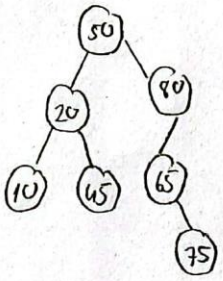
(5) Insert 65



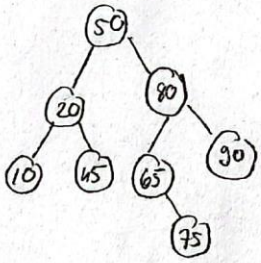
(6) hsert 75



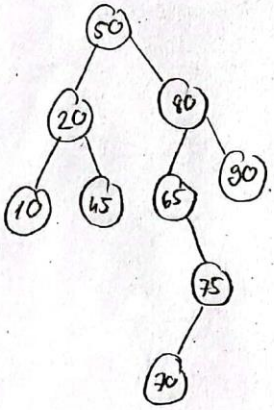
(7) hsert 45



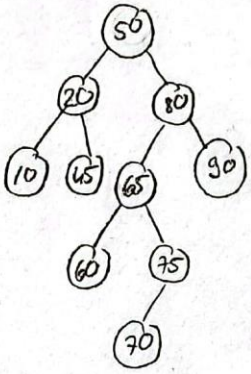
(8) hsert 90



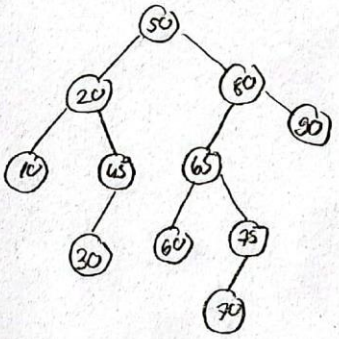
(9) hsert 70



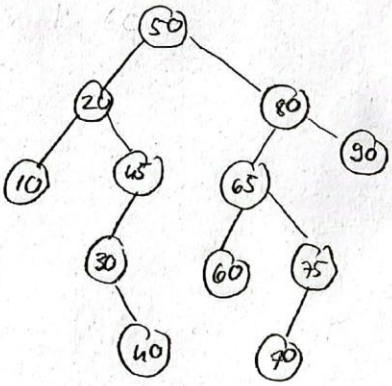
(10) hsert 60



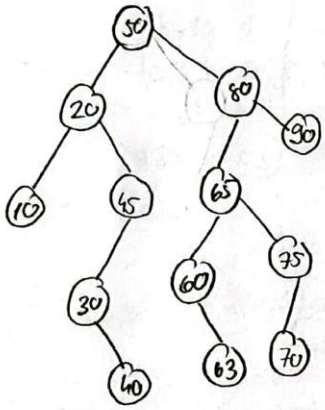
(11) hsert 30



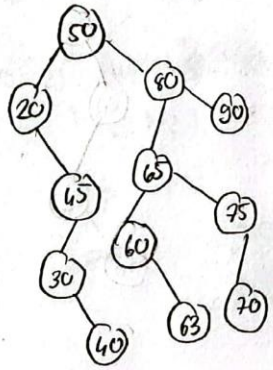
(12) hsert 40



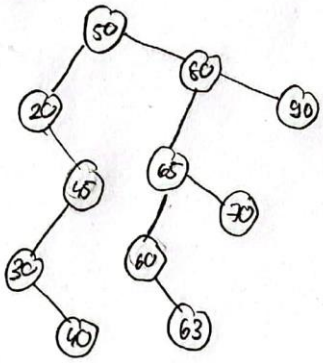
(13) Insert 63



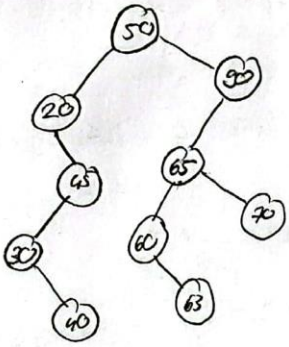
(14) delete 10



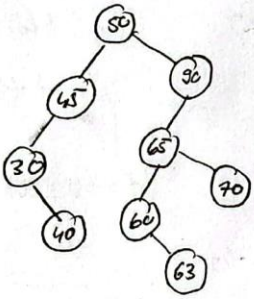
(15) delete 75



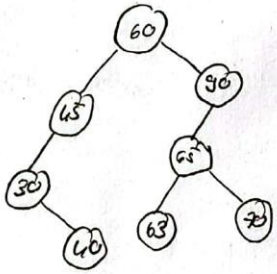
(16) delete 80



(17) delete 20

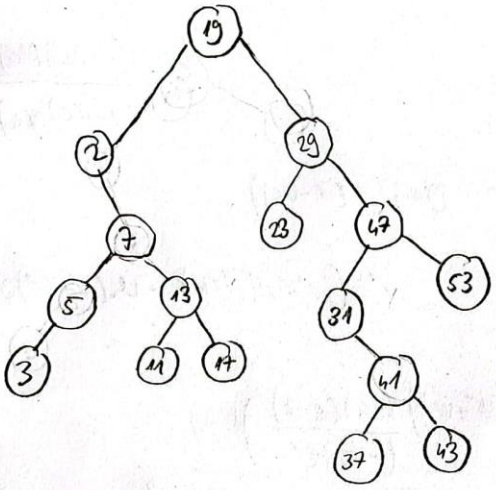


(18) delete 50



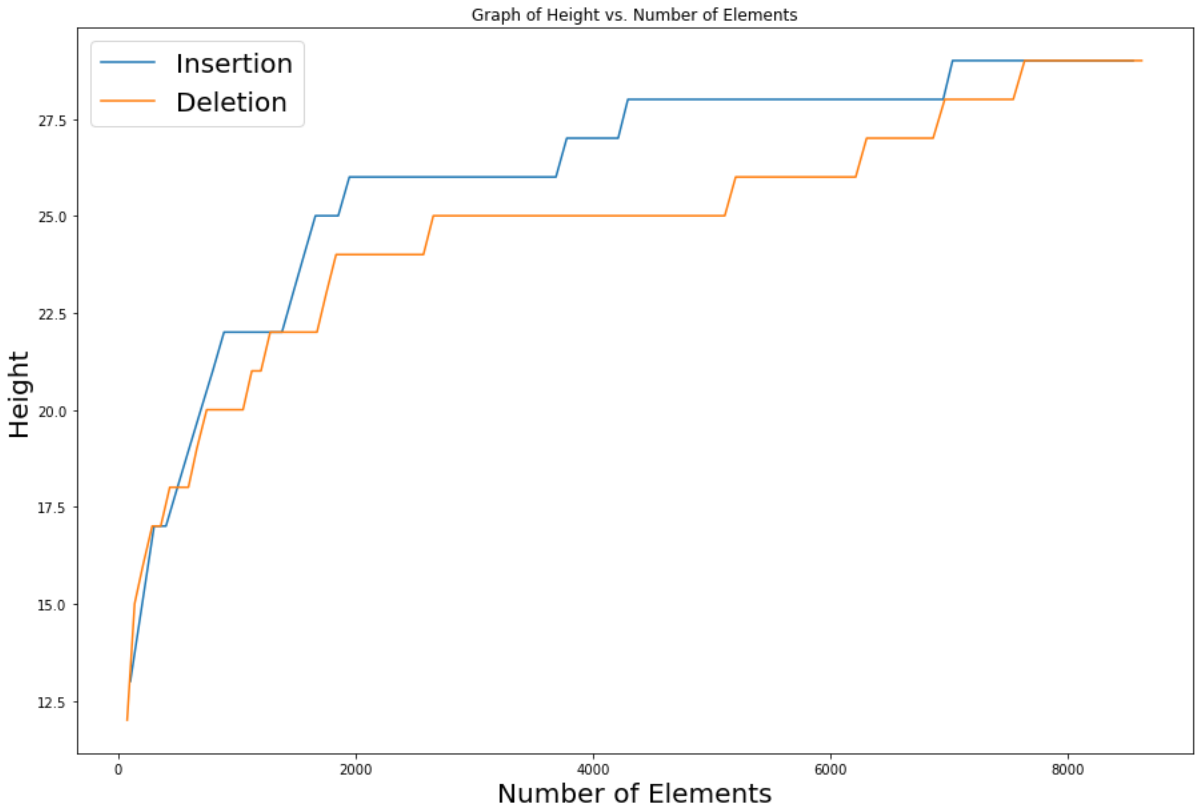
c.)

Binary Search Tree



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

Question 3.)



Analysis of Data

As it can be deduced from the graphs, both insertion and deletion show a logarithmic growth rate in average case. So, their run time in big-O notation is O(logn). The experimental results matched the theoretical ones as it is known that both insertion and deletion in binary trees take O(logn) time. This is because each method traverses the binary tree to find an insertion or a deletion point. In doing so, they skip one half of a subtree with each pass as the algorithms either go to the left child or to the right child based on the comparisons. Nevertheless, it works O(logn) because it is the average case and the elements in the array are created randomly. If the elements in the array were already sorted, then the height of the tree would have been “n” making the worst-case O(n) for both insertion and deletion. However, for the randomly created array the height is “logn” clearly parallel to the expected results.