CS202 Spring 2020-2021 Homework 2 Report

Title: Binary Search Trees

Name: Funda Tan

ID:21801861 Section: 3

Question 1

A)

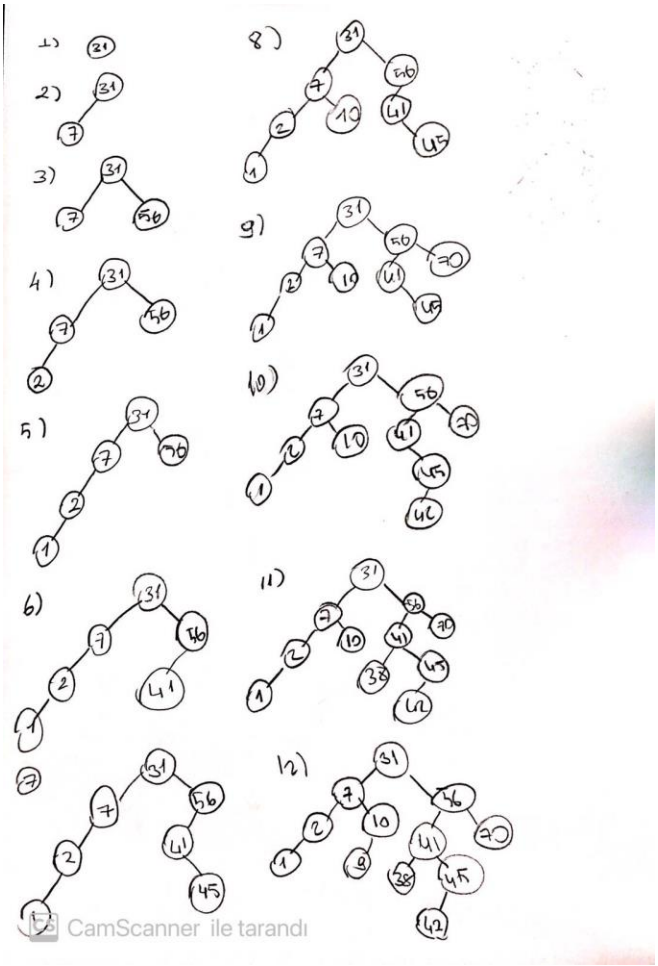
Prefix: / * A + B C D

Infix: A * B + C / D

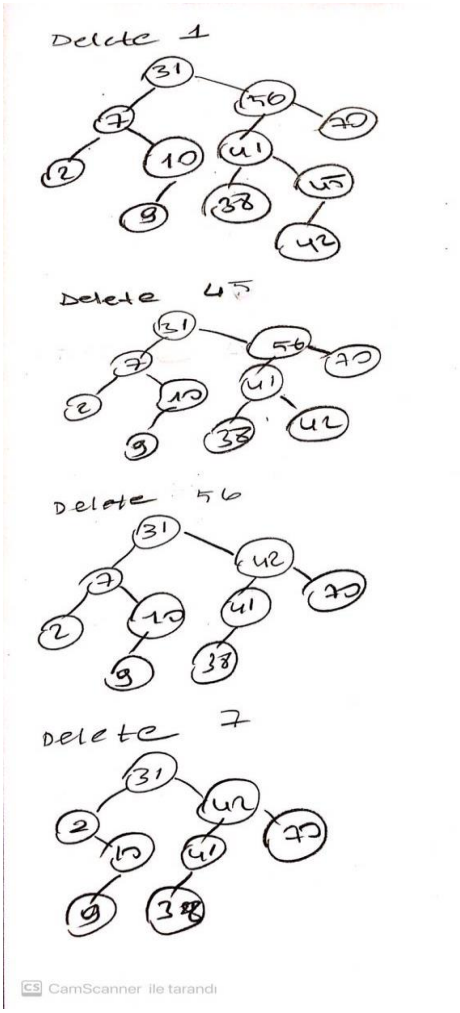
Postfix: B C + A * D /

B)

Insertion Operations

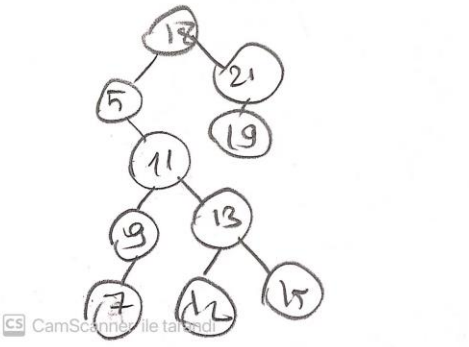


Deletion Operations



C)

Binary search tree from preorder traversal



Postorder traversal: 7 9 12 15 13 11 5 19 21 18

Question 3

levelorderTraverse:

In level order trace, I implemented a recursive algorithm. With recursion, it processes items in a level from left to right. Then, I iteratively merged these levels because it processes one level at a time. Level order traverse has O(n) time complexity because it processes every node once. Worst case is also O(n). Worst case time complexity is O(n). This algorithm cannot be implemented faster because to traverse the binary search tree, we need to process each and every node one by one.

span:

In span, I took two ends of the range and trace the binary tree recursively and update the counter. Span has O(n) time complexity. In worst case, span has O(n) time complexity because it has to visit all the nodes in the range and this range is fixed.

mirror:

In mirror, I go over the nodes recursively, then switched leftChildPtr and rightChildPtr using a temp pointer. Mirror has O(n) time complexity because it has to visit all of the nodes. Worst case time complexity is O(n). This algorithm cannot be implemented faster because to mirror the binary search tree, we need to process each and every node one by one.