Bilkent University - CS 202

Homework 1

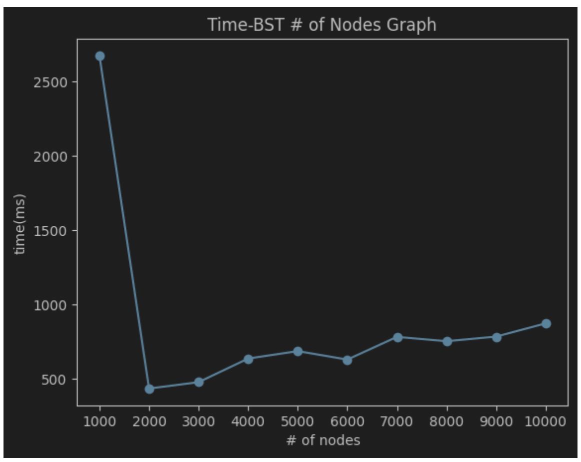
Alara Zeybek - 22102544
Section 3
04/03/2024

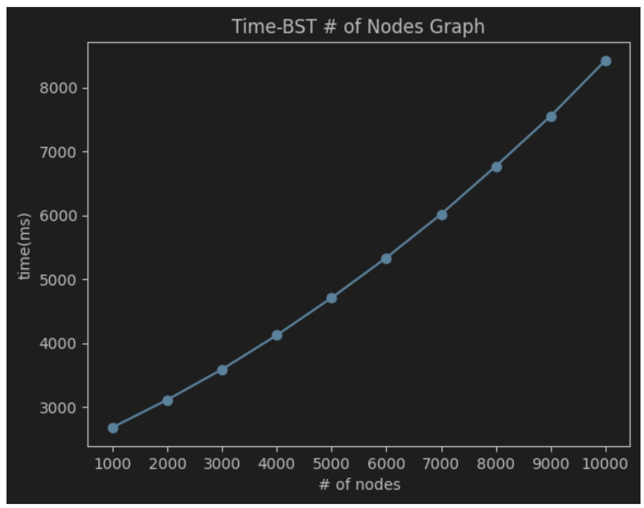
Question 3

The insertion of a node performs optimally when the binary tree is empty, as it only requires one pointer (root pointer) and takes O(1) time. However, the insertion procedure often involves recursive calls to locate the inserted object. As a result, each time we narrowed our search region by traveling right or left. This involves splitting the work in two. Insertion in a binary tree has an average time complexity of O(log(n). Since each recursive call divides the work by two. In a binary tree, the worst situation for insertion occurs when each node has just one child. This instance is comparable to the link list, so to get to the end. So, the worst complexity of insertion is O(n).

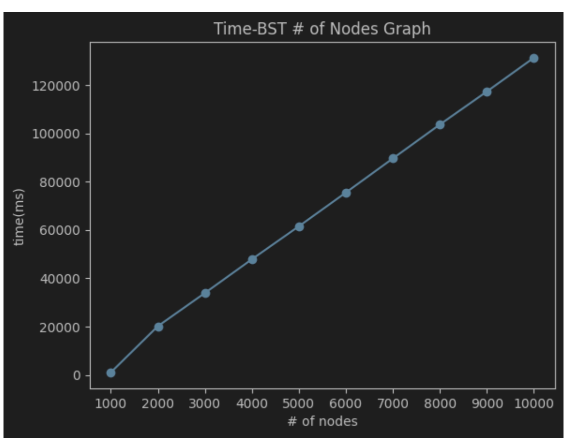
The plot of "BST height vs time" will most likely approximate a logarithmic function. This is typical behavior for a binary search tree, in which the height increases logarithmically with the number of nodes. The plot of elapsed time vs number of nodes may indicate an upward trend. A basic binary search tree insertion operation has an average time complexity of O(log n) for a balanced tree. However, factors such as implementation, hardware, and overhead can all have an impact on the actual elapsed time.

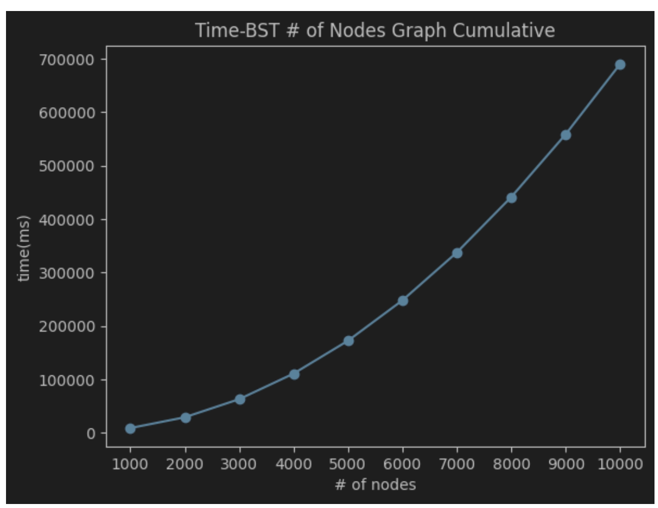
Random Generated Array:





Ordered array:





Q1)

a)

preorder : * + 11 ^ 4 2 - / 9 3 8

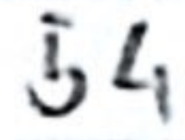
inorder : 11 + 4 ^ 2 * 9 / 3 - 8

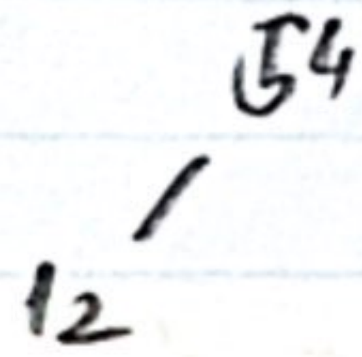
postorder: 11 4 2 ^ + 9 3 / 8 - *

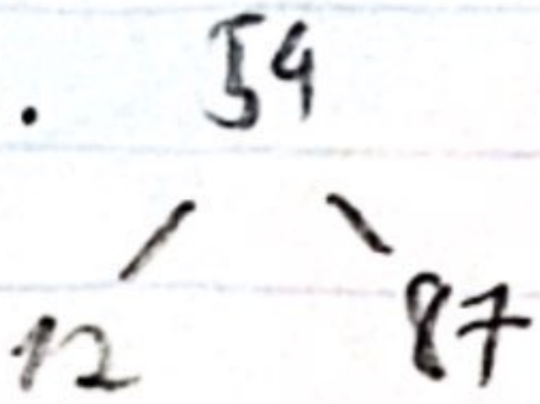
Calculating inorder = 11 + 16 * 9 / 3 - 8 = 11 + 48 - 8 = 51

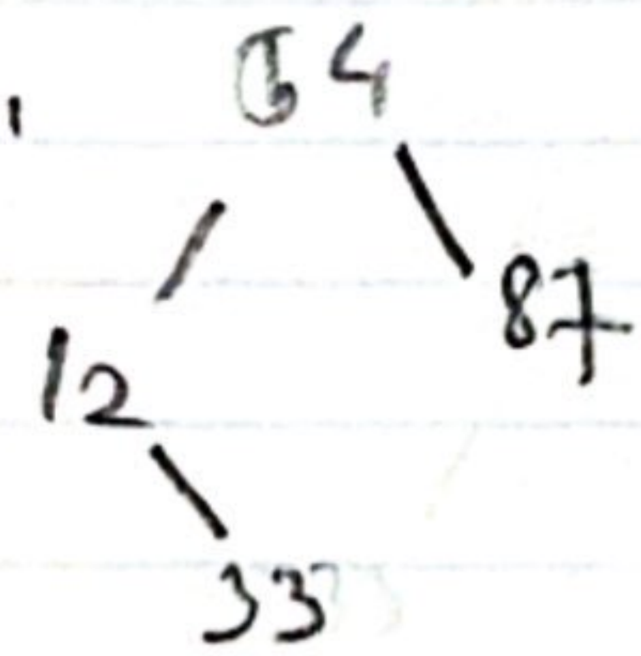
Scanned with CamScanner

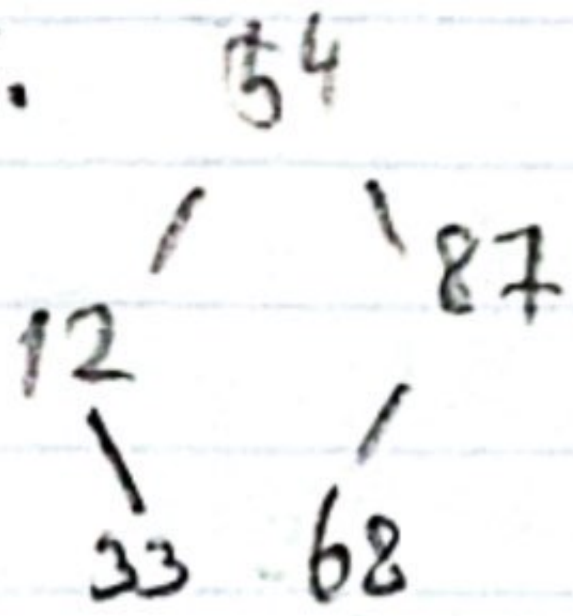
Insertion

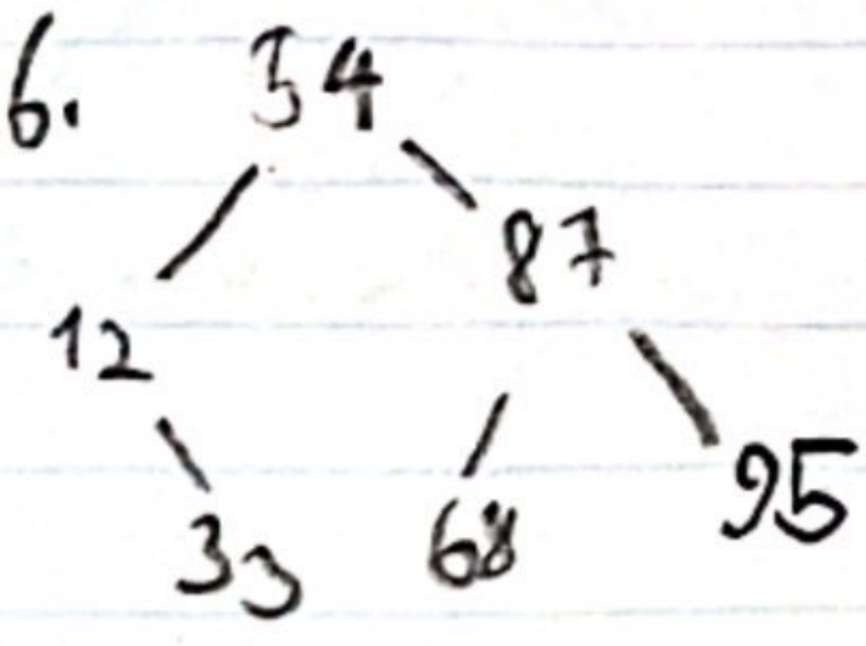
1. 

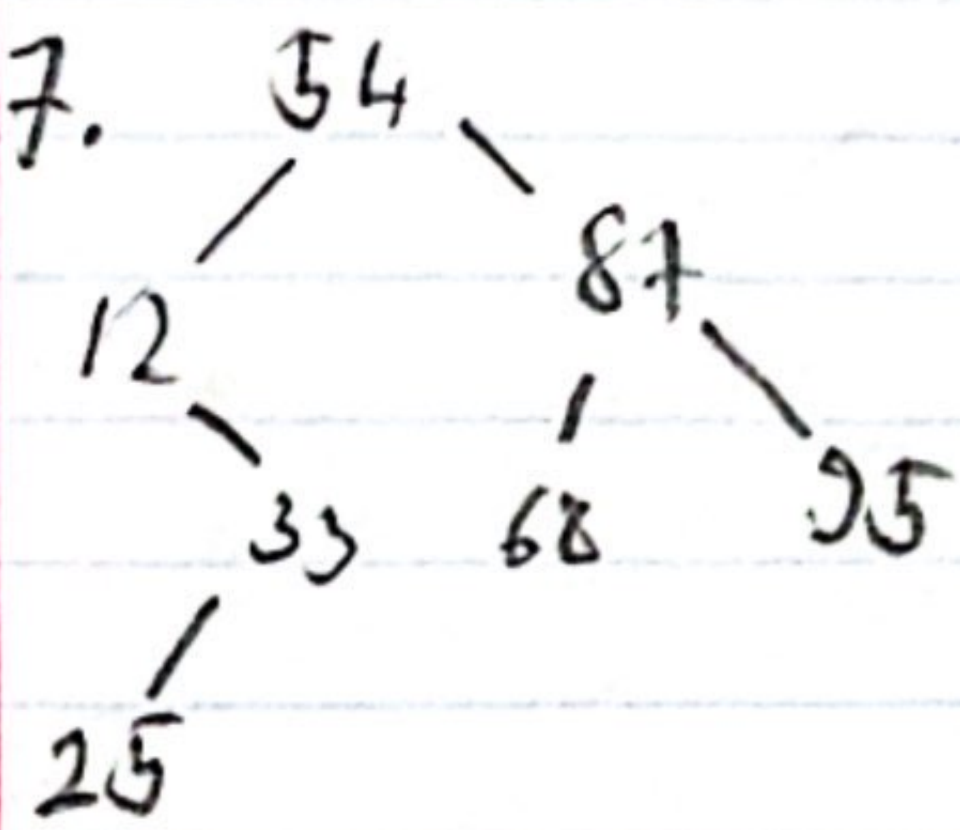
2. 

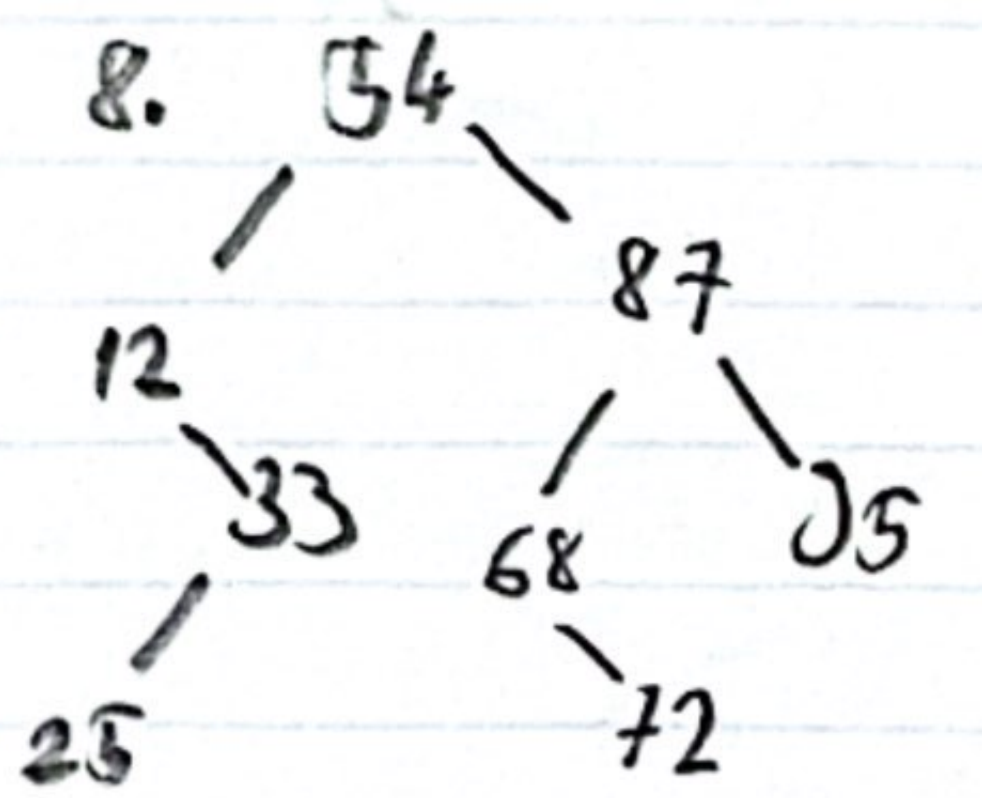
3. 

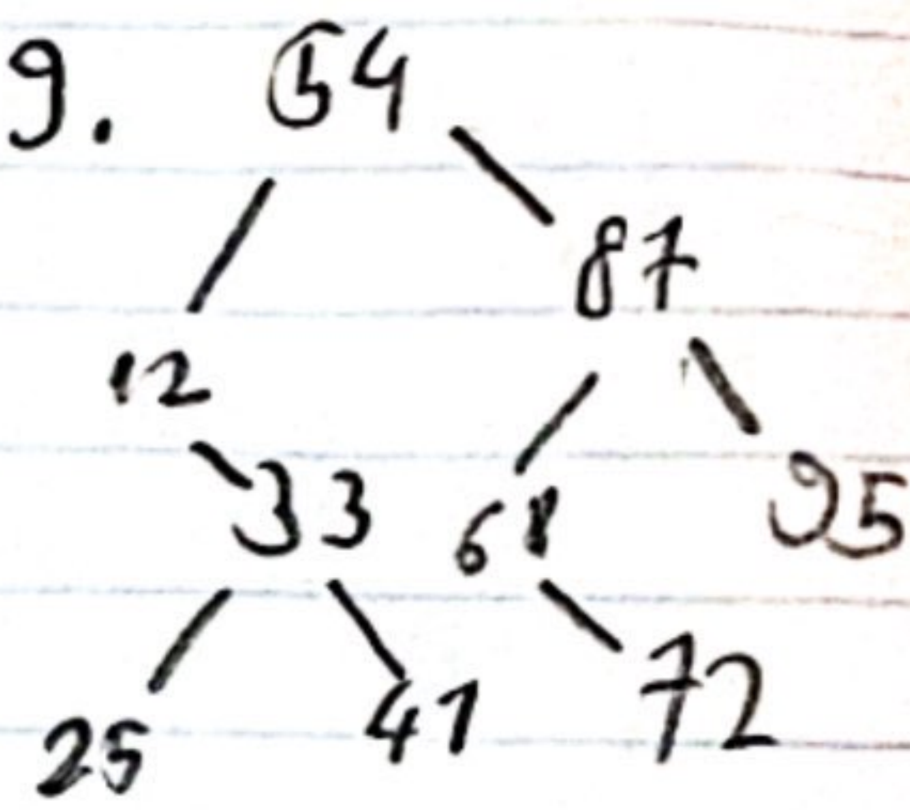
4. 

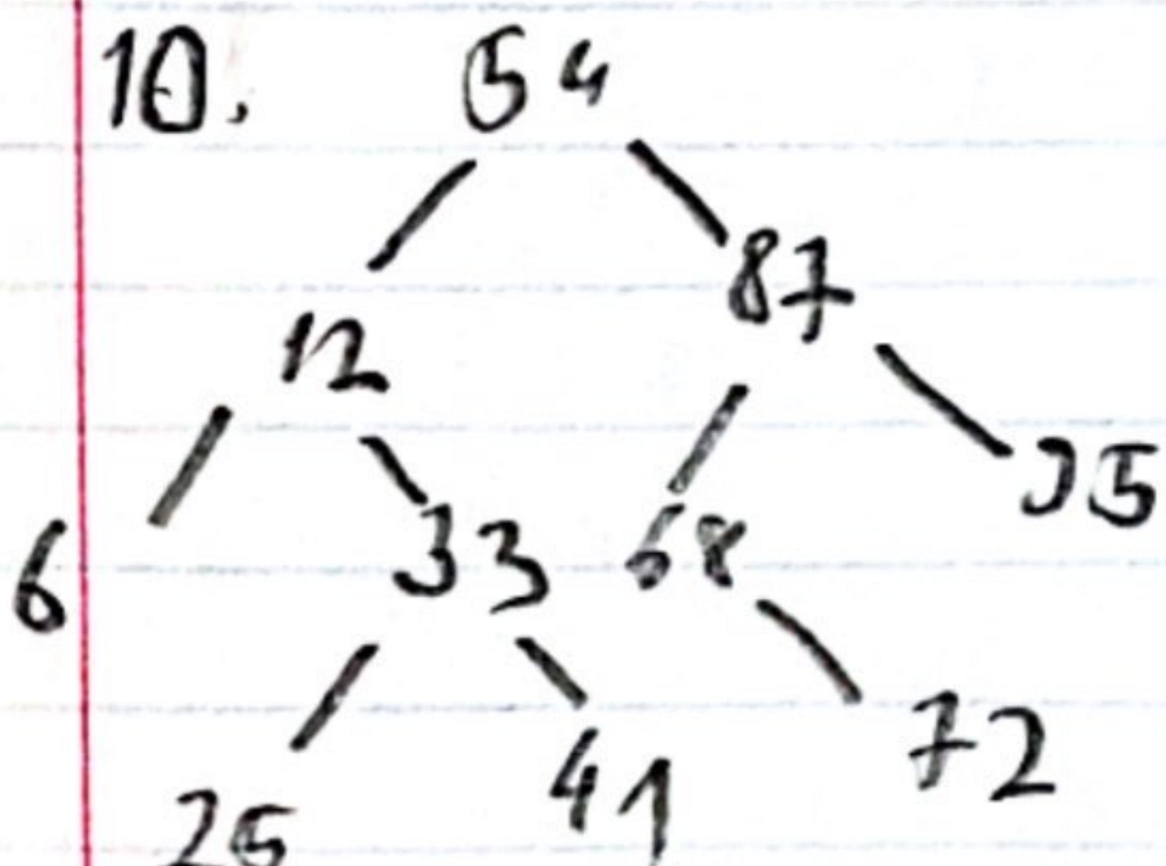
5. 

6. 

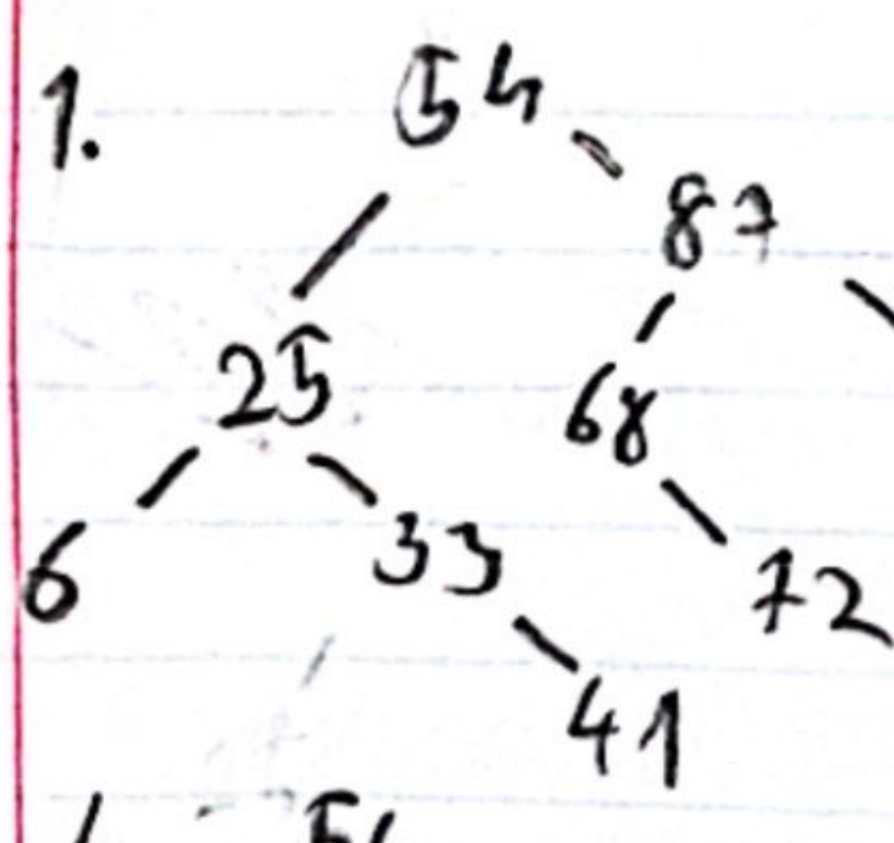
7. 

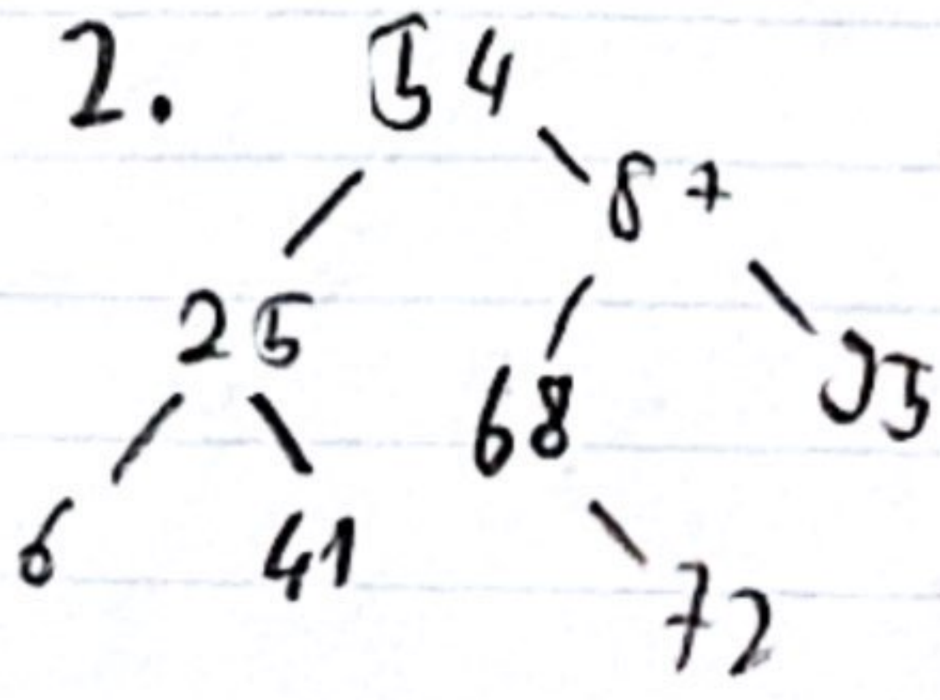
8. 

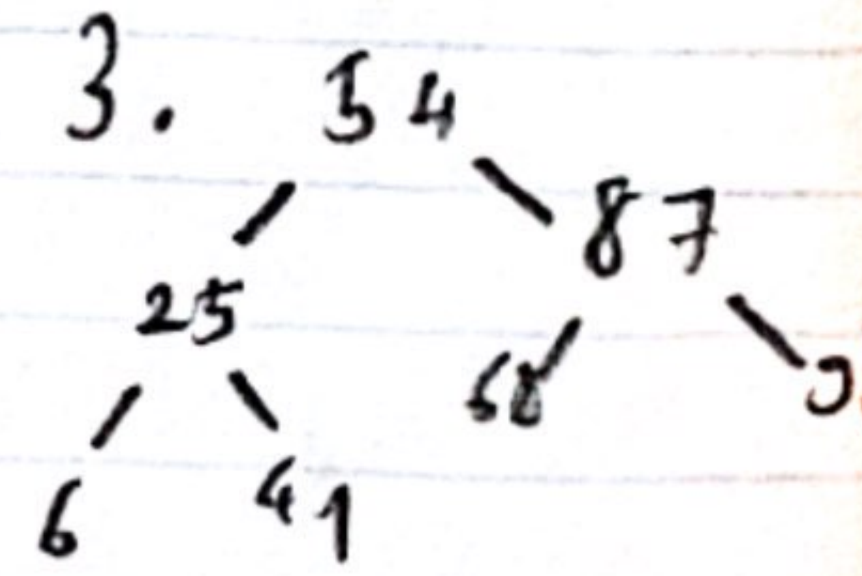
9. 

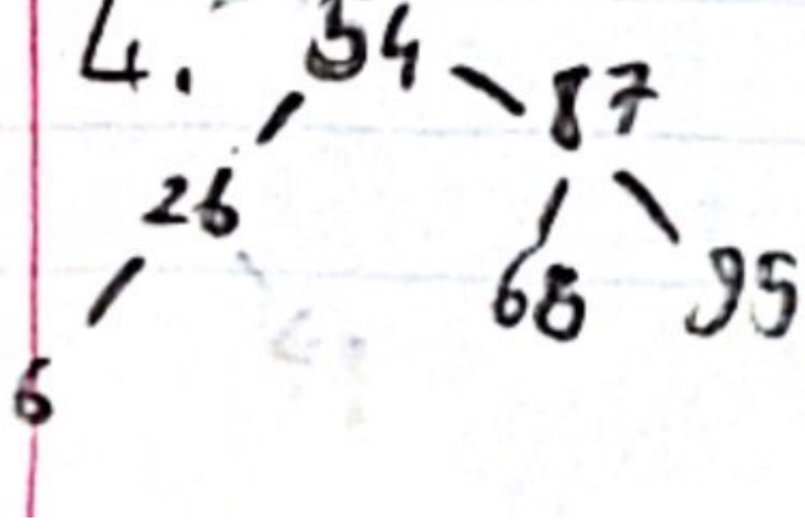
10. 

Deletion

1. 

2. 

3. 

4. 

c)

inorder = QRAMTJO