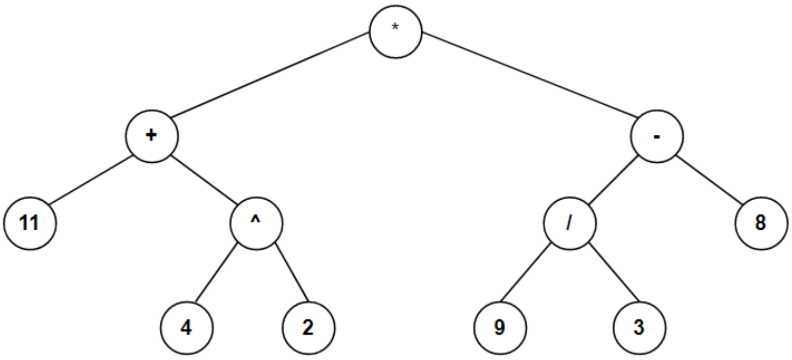
# CS202 - HW 1 - Report

Mehmet Akif Şahin - 22203673 - Section 3

26 February 2024

# Question 1

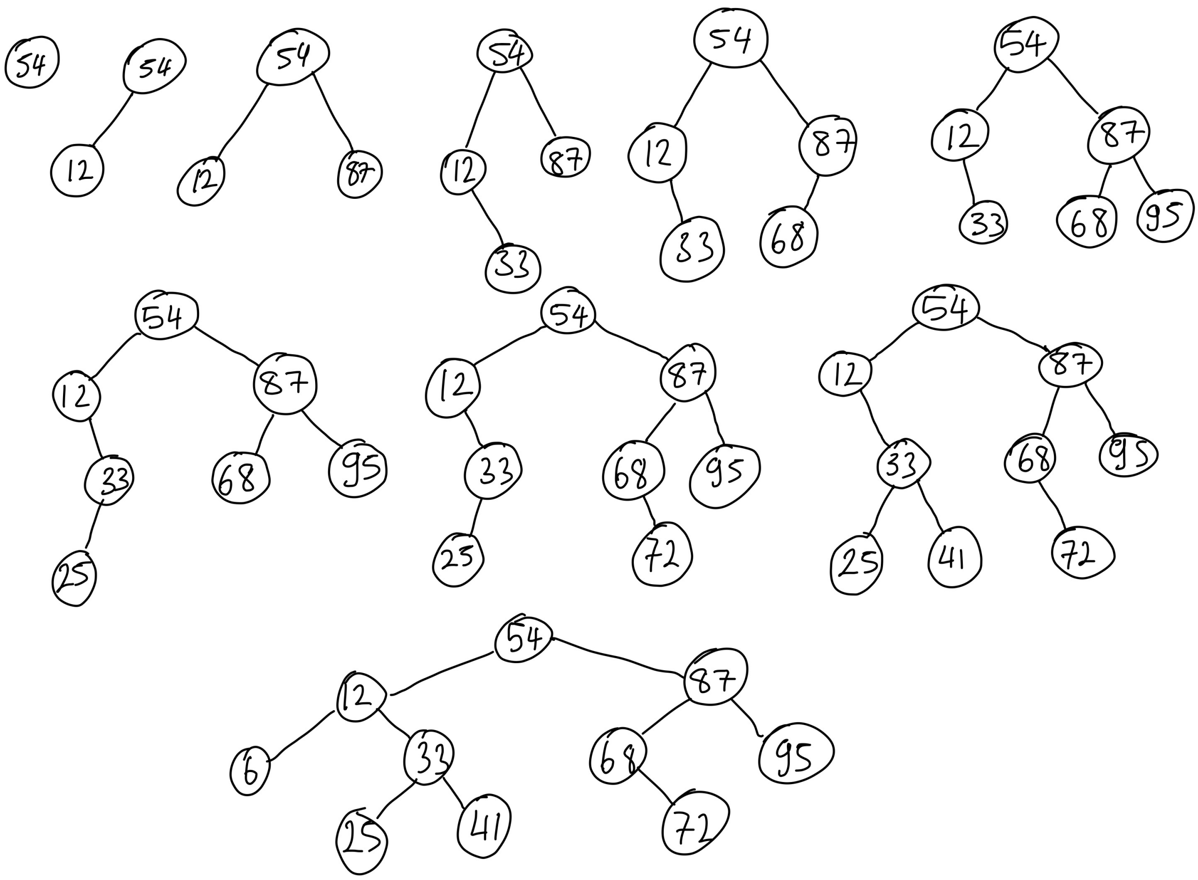
**a)** What are the preorder, inorder, and postorder traversals of the binary algebraic expression tree drawn below? Use the inorder traversal to compute the solution of the expression.



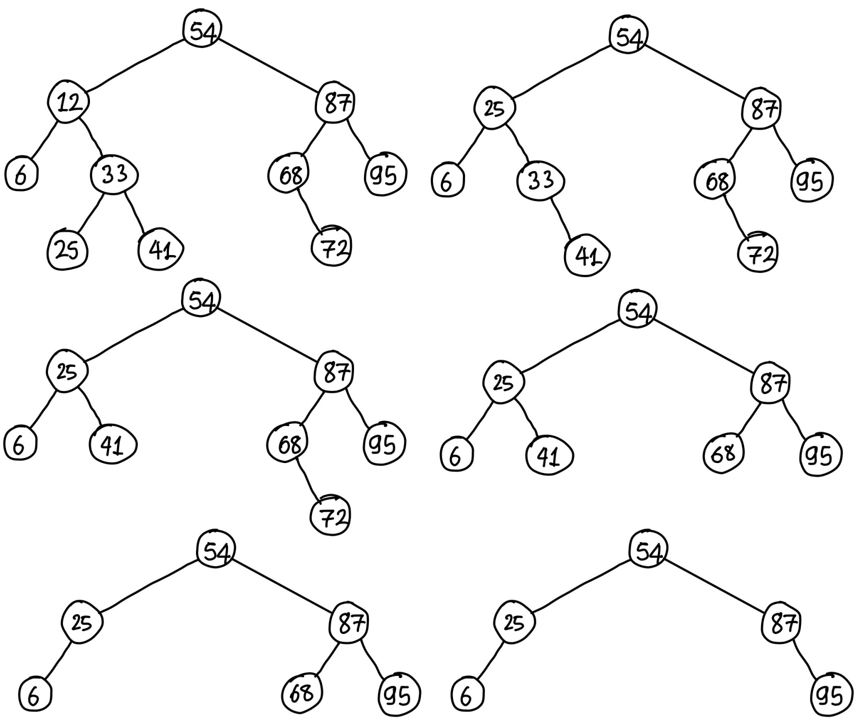

**Preorder Traversal:** $* + 11 \hat{} 4\ 2 - / 9\ 3\ 8$
**Inorder Traversal:** $11 + 4 \hat{} 2 * 9 / 3 - 8$
**Postorder Traversal:** $11\ 4\ 2 \hat{} + 9\ 3 / 8 - *$
**Solution** $(11 + 4^2) * (\frac{9}{3} - 8) = -135$

**b)** Insert 54, 12, 87, 33, 68, 95, 25, 72, 41, 6 into an empty binary search tree. Then delete 12, 33, 72, 41, 68. Show the tree after each insertion and deletion.

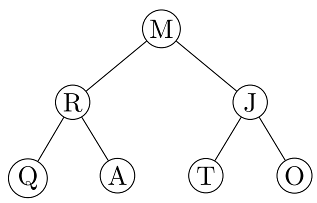
**Insertions** 54, 12, 87, 33, 68, 95, 25, 72, 41, 6



**Deletions** 12, 33, 72, 41, 68




**c)** The postorder traversal of a full binary tree is Q, A, R, T, O, J, M. What is its inorder traversal? Reconstruct the tree from its traversals and draw it.




**Inorder Traversal:** Q R A M T J O

# Question 3

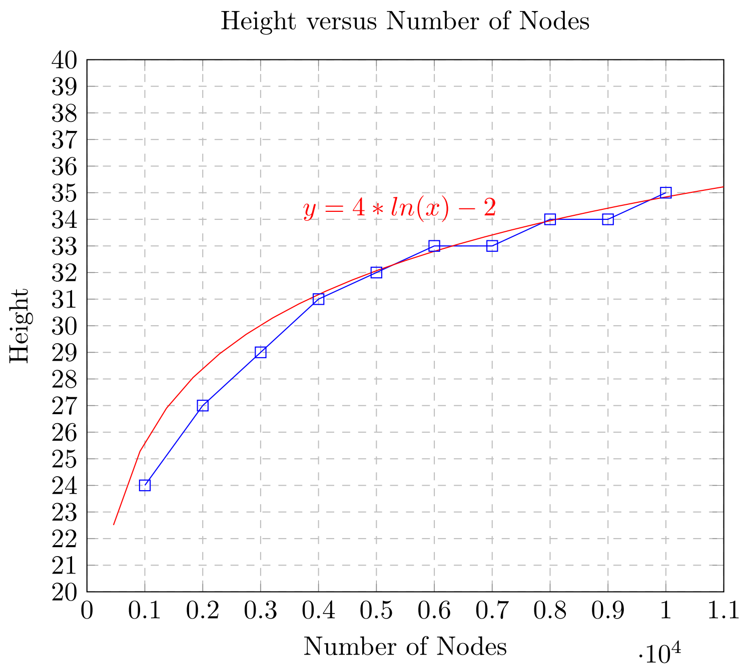



Figure 1: Binary Search Tree Height versus Number of Node Graph

In a binary search tree, on average, before the height increases, the number of nodes in the tree doubles. Thus, it is expected a plot of BST height versus the number of nodes to resemble a logarithmic curve. This can be seen in Figure 1.

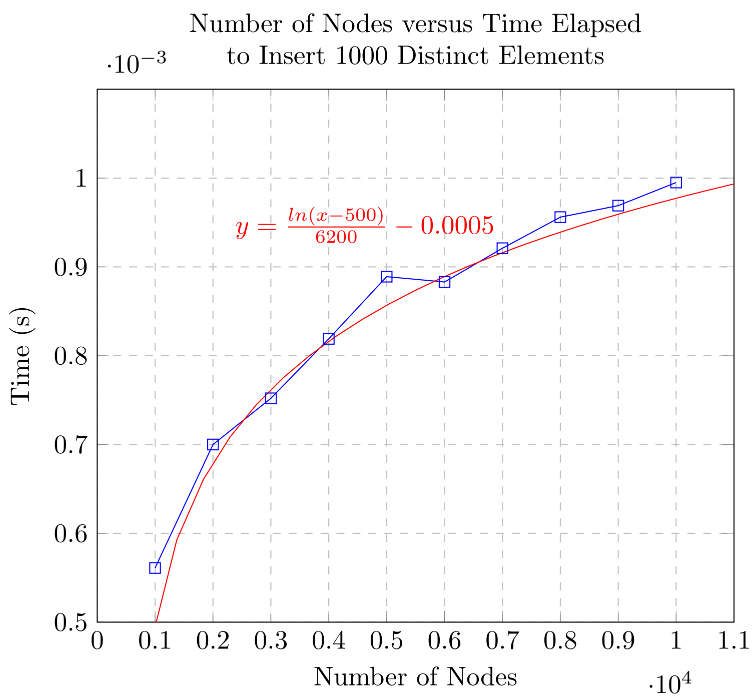



Figure 2: Binary Search Tree Time Elapsed to Insert 1000 Distinct Elements versus Number of Node Graph

The time complexity for insertion in a BST is O(log n) on average for random numbers as seen in Figure 2. However, if sorted integers inserted, the BST would become a linked list. And the time complexity for insertion becomes O(n). In this case, the plot of elapsed time versus the number of nodes would be a linear graph and this will be a huge inefficieny.