Author: Berke Kuzey Ardıç
ID: 22103340
Section: 001
Homework: 01

Question 1)

a) [5 points] 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: *, +, 11, ^, 4, 2, −, /, 9, 3, 8

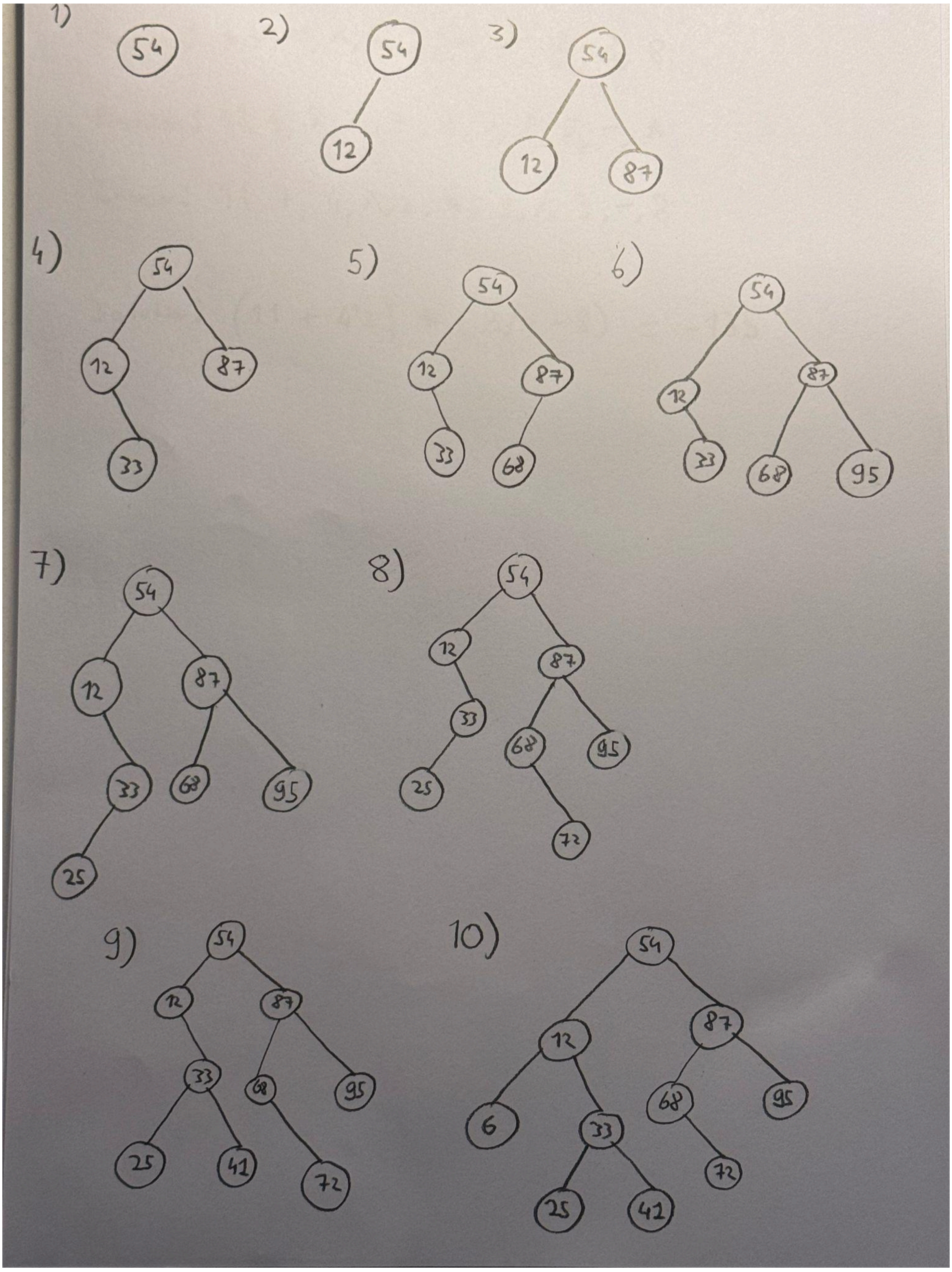
PostOrder: 11, 4, 2, ^, +, 9, 3, /, 8, −, *

Inorder: 11, +, 4, ^, 2, *, 9, /, 3, −, 8

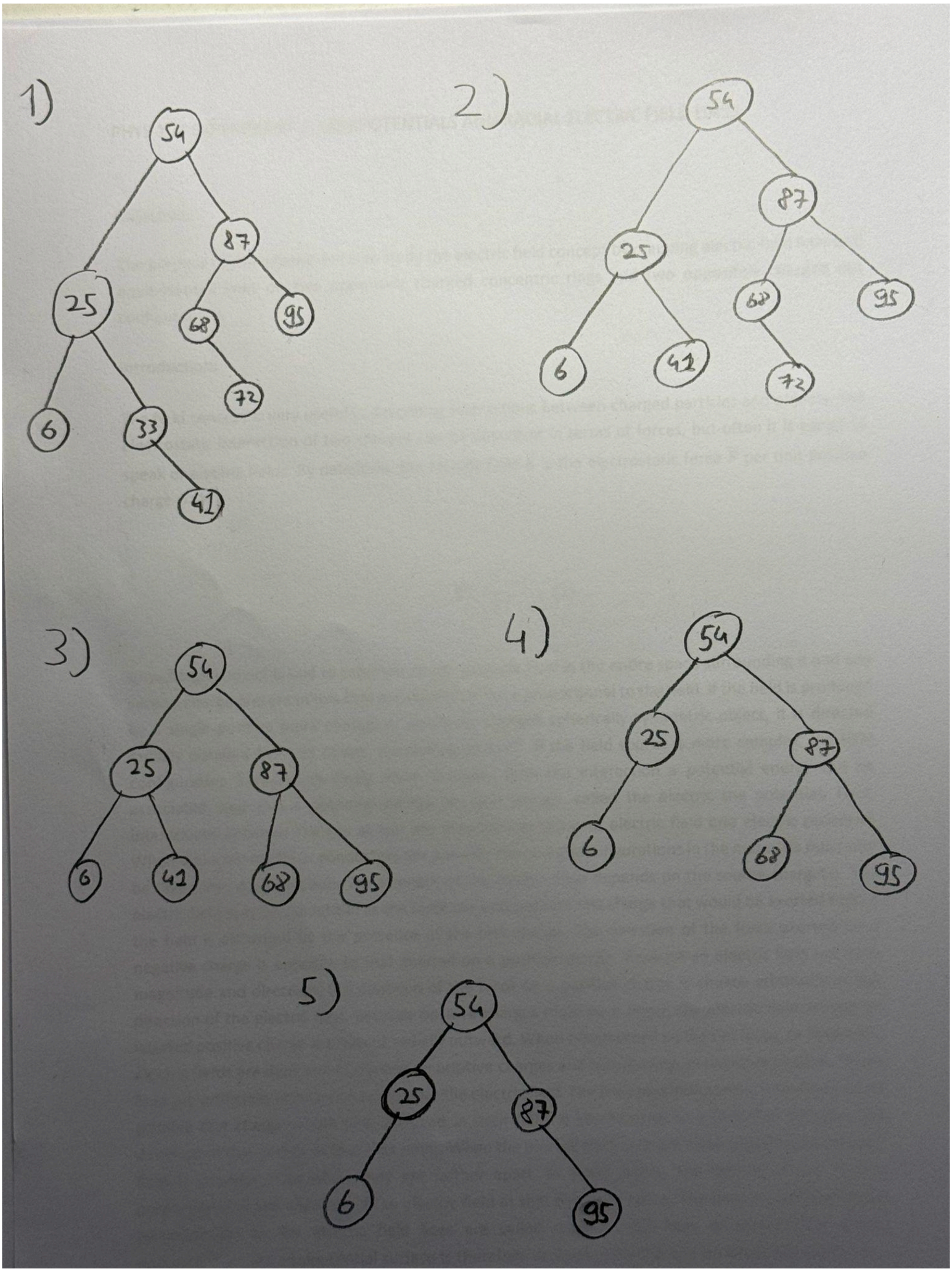
Solution: $(11 + 4^2) * (9/3 - 8) = -135$

b) [10 points] 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.

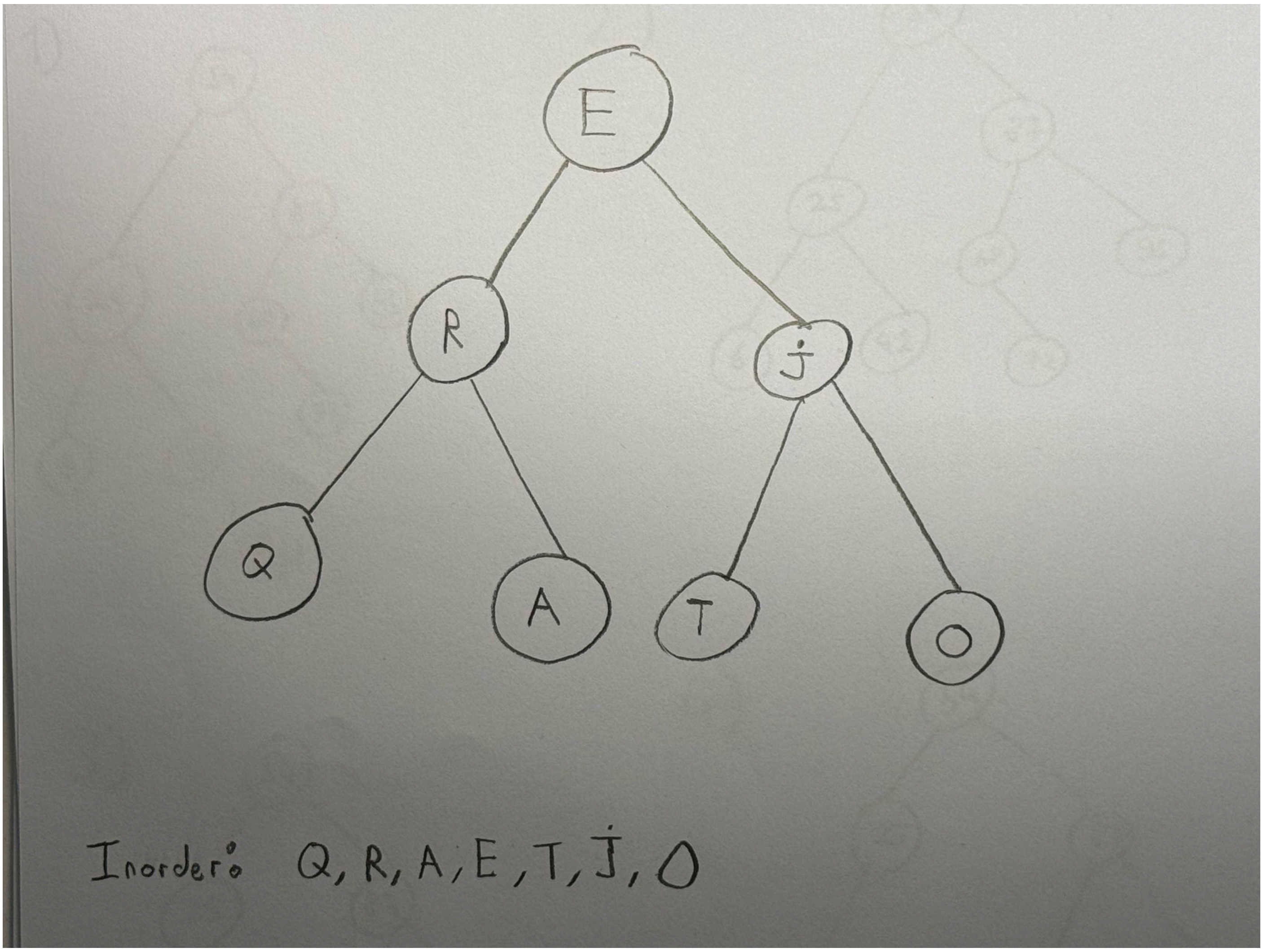
İnsertions:



Deletions:


1)
54
87
25
68
95
72
6
33
41
2)
54
87
25
68
95
6
41
72
3)
54
25
87
6
41
68
95
4)
54
25
87
6
68
95
5)
54
25
87
6
95

c) [5 points] The postorder traversal of a full binary tree is Q, A, R, T, O, J, M, R, E. What is its inorder traversal? Reconstruct the tree from its traversals and draw it.



Question 3)

1. The function BST height versus the number of nodes in the BST graph behaves between logarithmic and linear structure but it looks like a logarithmic function rather than a linear structure. Because as the number of nodes grows, height doesn't grow with the same rate, instead it grows slowly. This situation is parallel to expected behavior of BST under ideal conditions because the height of a well balanced tree tends to grow in a logarithmic order. In this situation the best case is a perfectly balanced tree in which the left and right subtrees height differ by at most one and the worst case is a completely unbalanced tree which will be actually a linked list. In the best case scenario insertion time is O(log(n)) and this makes the operations of search, insertion and deletion faster. In the worst case scenario the insertion time O(n) is like a linkedlist. The reason this graph has a behavior between a balanced tree and unbalanced one is the numbers are randomly generated. In conclusion for the first plot the behavior is similar to the theoretical expectations.
2. The graph shows the elapsed time grows slower than the number of nodes inserted to the tree. Again the behavior is between a logarithmic and linear structure but it is closer to a logarithmic structure. In this situation the best case time complexity of insertion is O(h) where h represents the tree height and as we mentioned in the first part height tends to grow in a logarithmic manner depending on the number nodes in a perfectly balanced tree and the worst case is O(n) where n represents the number of the nodes in the tree and it occurs with a completely unbalanced tree. Again since we inserted the numbers in a random manner the graph has a behavior between a balanced tree and unbalanced one. Although if we insert sorted integers the insertion time would significantly change because sorted integers performs the completely unbalanced tree which is the worst case of insertion time complexity, because each new insertion adds a level to the tree, requiring a traversal through all the previous nodes.