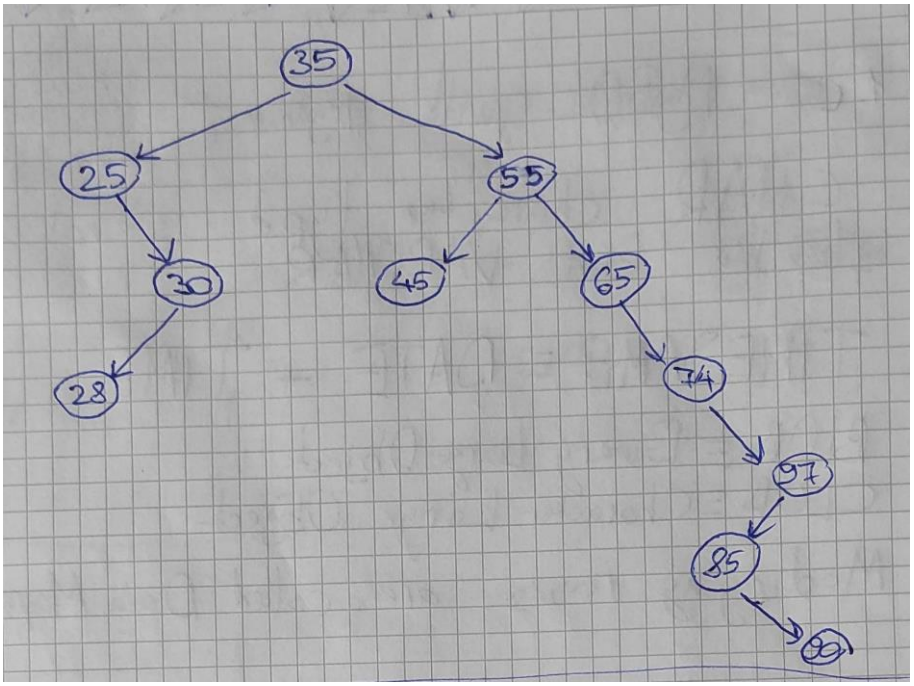
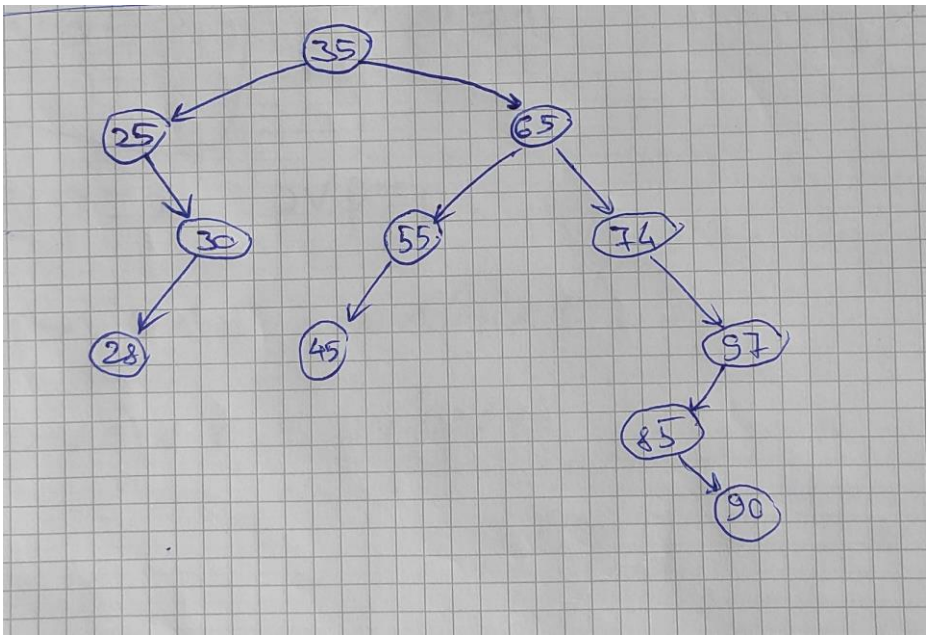
Report

1. Short answer is 7

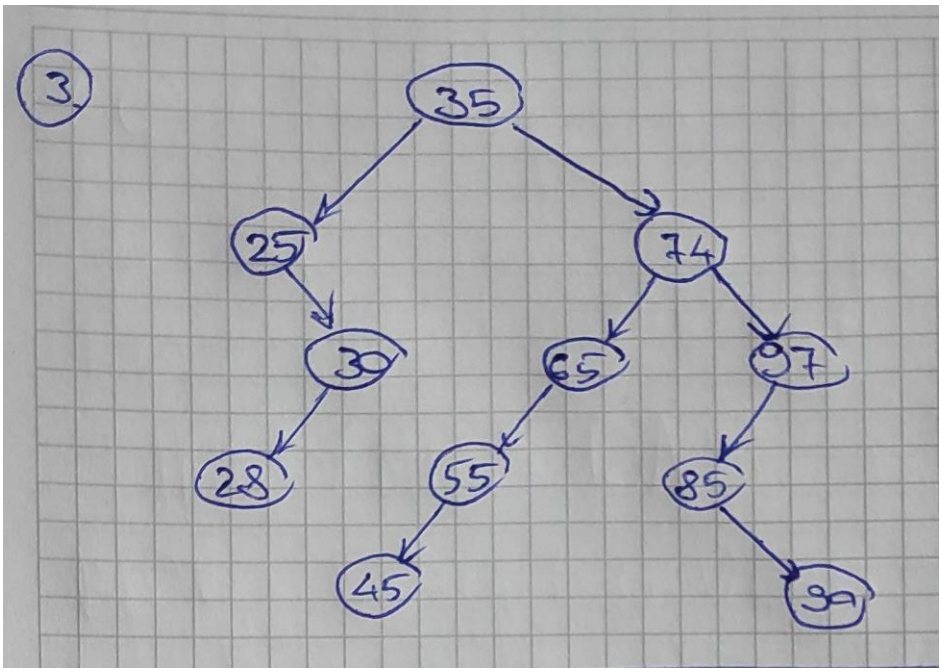
Step 1: Left Rotate 45



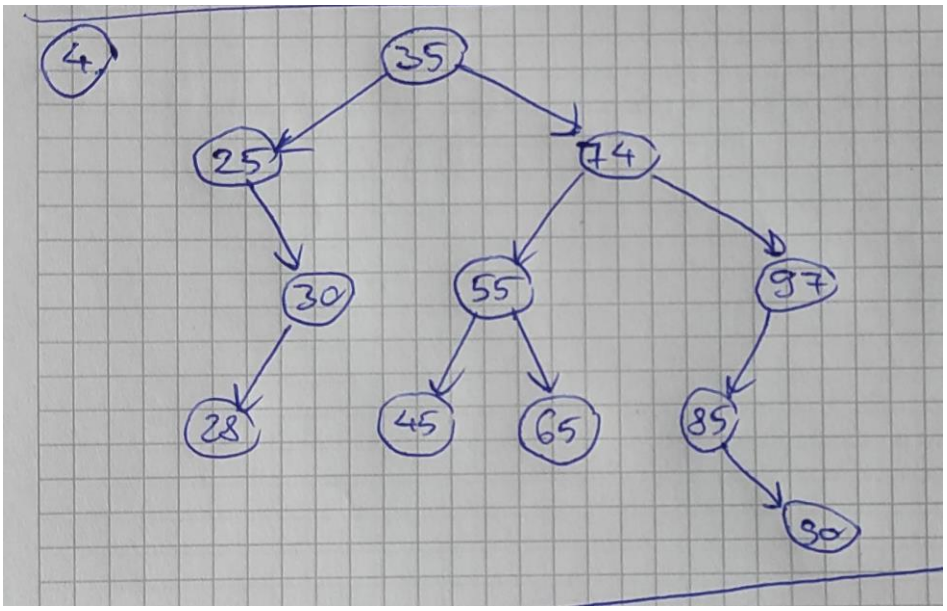
Step 2: Left Rotate 55



Step 3: Left Rotate 65



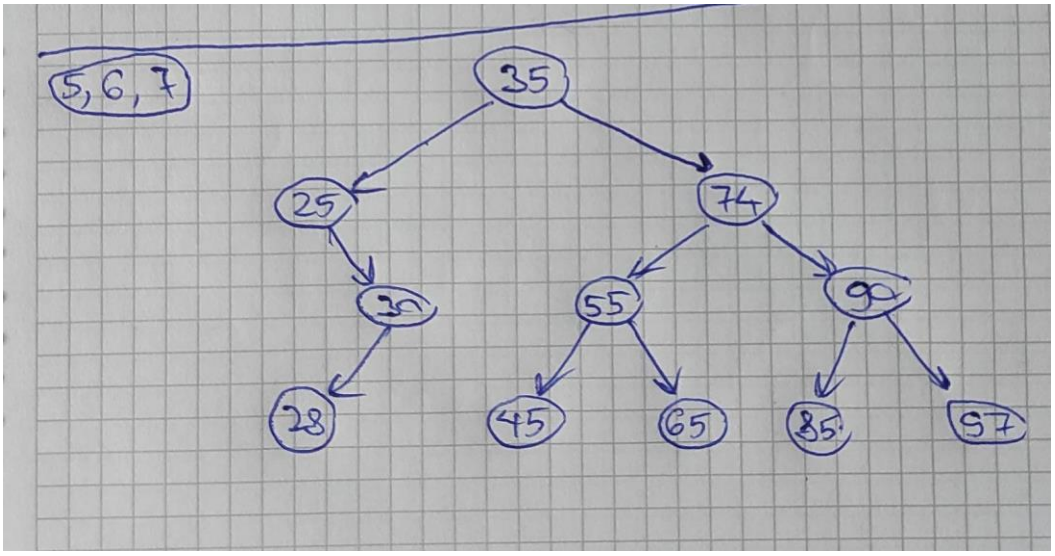
Step 4: Right Rotate 65



Step 5: Right Rotate 97

Step 6: Right Rotate 97

Step 7: Left Rotate 85



2. While a standard binary search tree is not self balancing due to insertion order, it will have various time complexities. However red black tree rebalances itself and provides same time complexity.

3. Red black tree has height smaller or equal to 2 log(n+1), traversing tree has O(2log(n+1)) upper bound which means O(log(n)). Insertion operation of red black tree has same meaning as insertion to binary search tree with O(log(n)) height. Afterwards necessary rotating and reorganizing operations are needed which have O(1) complexity. Since O(log(n)) + O(1) = O(log(n)), asymptotic upper bound of insertion for average case is O(log(n)). While in worst case also red black tree has 2 log(n) height it will be no different than the average case. Thus, worst case has same complexity with average case which is O(log(n)).

4.

We should store 5 size values for genres at first. Then we have to store genre of the i'th node either. Any node we choose will contain information of number of video games in same genre. Inertion and rotation operations can be done. By

inorder traversal, if we increase any node’s left child’s genre size it will provide us the node.