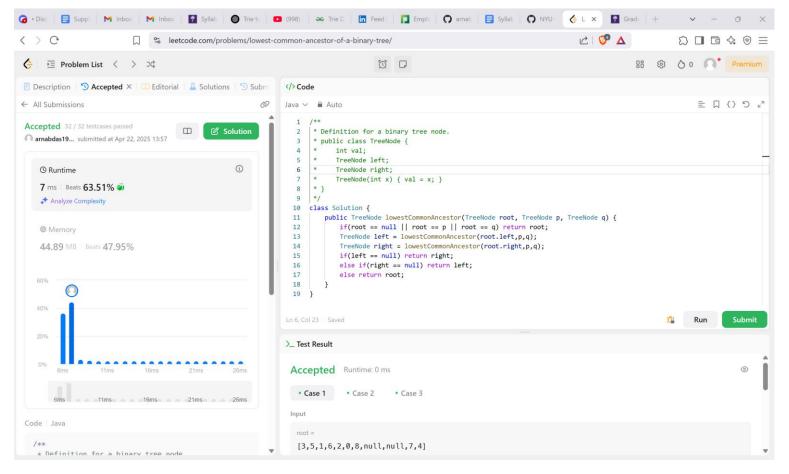
Assignment-6

```
Question 1.
/**
 * Definition for a binary tree node.
  public class TreeNode {
       int val;
       TreeNode left;
       TreeNode right;
       TreeNode(int x) { val = x; }
 * }
 */
class Solution {
    public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {
        if(root == null || root == p || root == q) return root;
        TreeNode left = lowestCommonAncestor(root.left,p,q);
        TreeNode right = lowestCommonAncestor(root.right,p,q);
        if(left == null) return right;
        else if(right == null) return left;
        else return root;
    }
}
```



Code Java

class Solution {

ublic int[] tonKFrequent(int[] nums int k) {

```
class Solution {
                   public int[] topKFrequent(int[] nums, int k) {
                                        HashMap<Integer, Integer> map = new HashMap<>();
                                        for(int num : nums){
                                                           map.put(num,map.getOrDefault(num,0)+1);
                                        }
                                        PriorityQueue<Integer> pq = new PriorityQueue<>((a,b) -> map.get(b) - map.get(a));
                                        pq.addAll(map.keySet());
                                        int [] result = new int[k];
                                        for(int i =0;i<k;i++){</pre>
                                                           result[i] = pq.poll();
                                        }
                                        return result;
                    }
}
                                Suppl M Inbox 
                                                                                 \begin{tabular}{lll} $\cong$ & leetcode.com/problems/top-k-frequent-elements/?envType=problem-list-v2&envId=heap-priority-queue \\ \end{tabular}
                                                                                                                                                                                                                                                                                                                                                                                                C | 👽 🛕
                                                                                                                                                                                                                                                                                                                                                                                                                                                                    BB ⊗ O O Premium
        Heap (Priority Queue)
        ■ Description
Submed X
Description
Description
Submed X
Description
Descri
                                                                                                                                                                                        </>Code
        ← All Submissions

    Auto

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 E □ () □ =
                                                                                                                                                                                                 1 class Solution (
         Accepted 21 / 21 testcases passed
                                                                                                                                                                                                                    public int[] topKFrequent(int[] nums, int k) {
         arnabdas19... submitted at Apr 22, 2025 14:02
                                                                                                                                                                                                                             HashMap<Integer, Integer> map = new HashMap<>();
for(int num : nums){
                                                                                                                                                                                                                                         map.put(num,map.getOrDefault(num,0)+1);
                                                                                                                                                        (i)
                   © Runtime
                                                                                                                                                                                                                               PriorityQueue<Integer> pq = new PriorityQueue<>((a,b) -> map.get(b) - map.get(a));
                    13 ms | Beats 76.53% 🞳
                                                                                                                                                                                                                              pq.addAll(map.keySet());
int [] result = new int[k];
for(int i =0;i<k;i++){</pre>
                   Analyze Complexity
                                                                                                                                                                                               10
                                                                                                                                                                                                                                        result[i] = pq.poll();
                                                                                                                                                                                               12
                   @ Memory
                                                                                                                                                                                                                               return result;
                   48.80 MB | Beats 53.26% 🔊
                                                                                                                                                                                          Ln 10, Col 20 Saved
                                                                                                                                                                                          >_ Test Result
                                                                                                                                                                                             Accepted Runtime: 1 ms
                                          6ms 11ms 16ms
                                                                                                                                                                                              • Case 1 • Case 2
```

Input

[1,1,1,2,2,3]