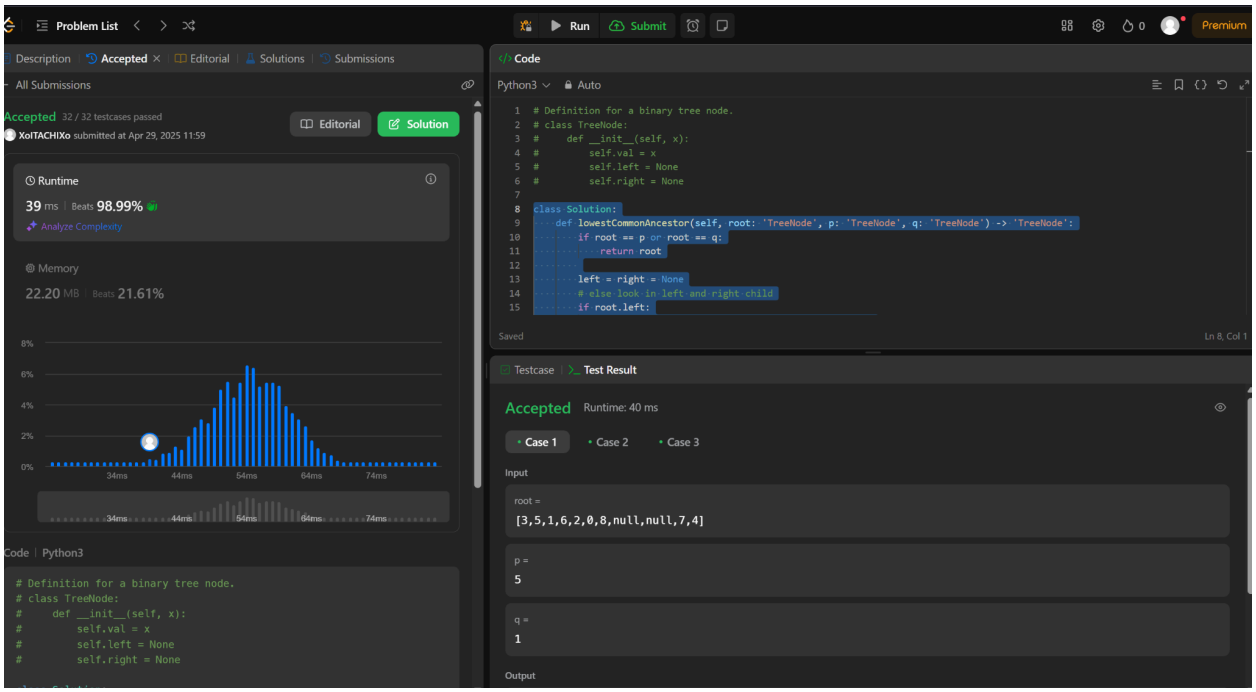
LeetCode Bootcamp Week 6

```python
class Solution:
    def lowestCommonAncestor(self, root: 'TreeNode', p: 'TreeNode', q: 'TreeNode') -> 'TreeNode':
        if root == p or root == q:
            return root

        left = right = None
        # else look in left and right child
        if root.left:
            left = self.lowestCommonAncestor(root.left, p, q)
        if root.right:
            right = self.lowestCommonAncestor(root.right, p, q)

        if left and right:
            return root
        else:

            return left or right
```



```python
class Solution:
    def topKFrequent(self, nums: List[int], k: int) -> List[int]:
        count = Counter(nums)
        sort_count_keys = sorted(count.keys(), key=lambda x: count[x])
        return sort_count_keys[-k:]
```