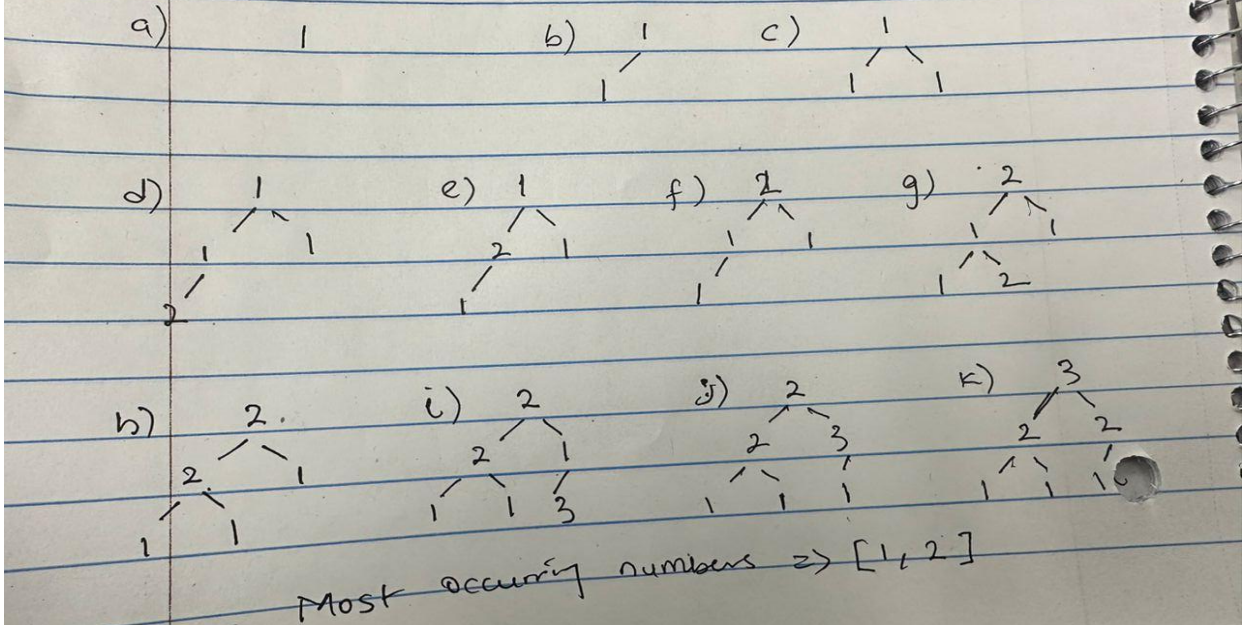
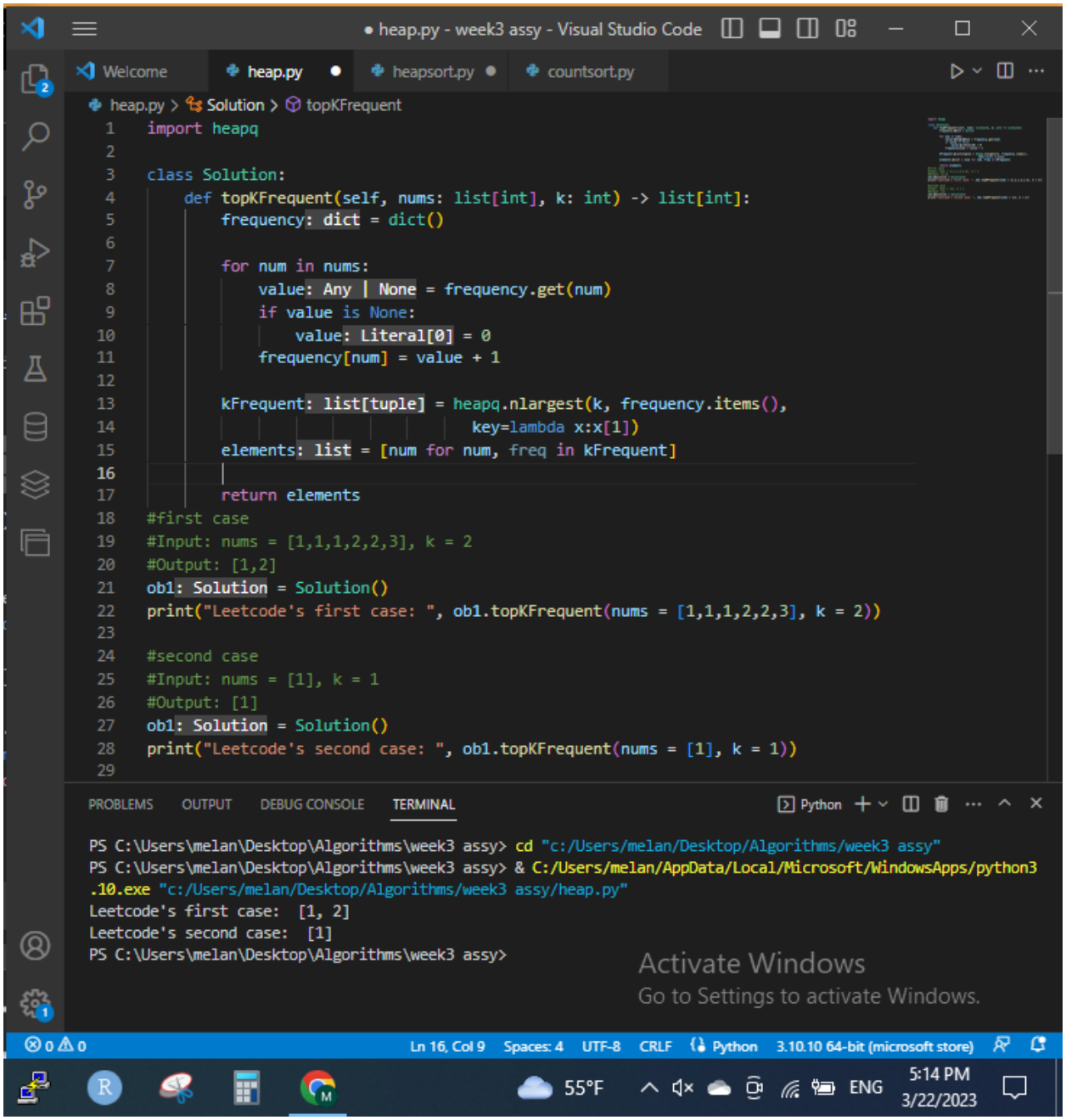
Heap

1 1 1 2 2 3 (max heap)



Most occurring numbers => [1, 2]



CODE

```
import heapq

class Solution:
    def topKFrequent(self, nums: list[int], k: int) -> list[int]:
        frequency = dict()

        for num in nums:
```

```
            value = frequency.get(num)
            if value is None:
                value = 0
            frequency[num] = value + 1

        kFrequent = heapq.nlargest(k, frequency.items(), key=lambda x:x[1])
        elements = [num for num, freq in kFrequent]

        return elements
#first case
#Input: nums = [1,1,1,2,2,3], k = 2
#Output: [1,2]
ob1 = Solution()
print("Leetcode's first case: ", ob1.topKFrequent(nums = [1,1,1,2,2,3], k = 2))

#second case
#Input: nums = [1], k = 1
#Output: [1]
ob1 = Solution()
print("Leetcode's second case: ", ob1.topKFrequent(nums = [1], k = 1))
```