

Heap

1 1 1 2 2 3 (max heap)

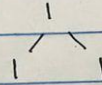
a)

1

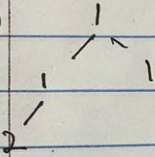
b)



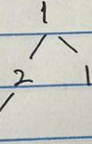
c)



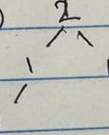
d)



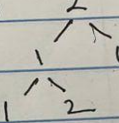
e)



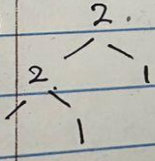
f)



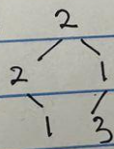
g)



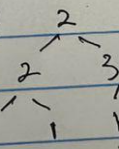
h)



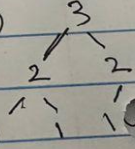
i)



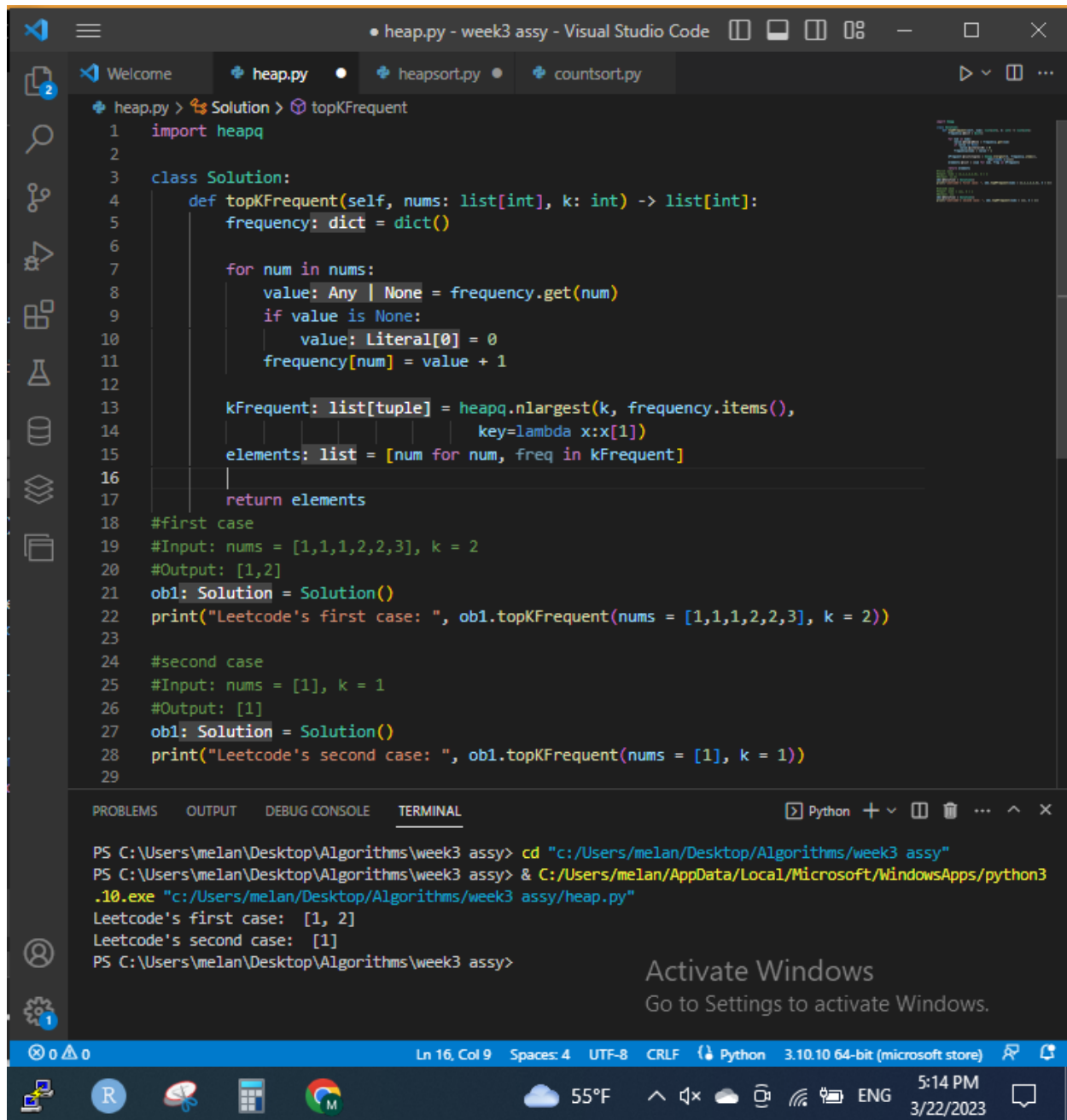
j)



k)



Most occurring numbers $\Rightarrow [1, 2]$



```
heap.py - week3 assy - Visual Studio Code
Welcome | heap.py | heap.sort.py | countsort.py
heap.py > Solution > topKFrequent
1 import heapq
2
3 class Solution:
4     def topKFrequent(self, nums: list[int], k: int) -> list[int]:
5         frequency: dict = dict()
6
7         for num in nums:
8             value: Any | None = frequency.get(num)
9             if value is None:
10                 value: Literal[0] = 0
11                 frequency[num] = value + 1
12
13         kFrequency: list[tuple] = heapq.nlargest(k, frequency.items(),
14                                             key=lambda x:x[1])
15         elements: list = [num for num, freq in kFrequency]
16
17         return elements
18
19 #first case
20 #Input: nums = [1,1,1,2,2,3], k = 2
21 #Output: [1,2]
22 ob1: Solution = Solution()
23 print("Leetcode's first case: ", ob1.topKFrequent(nums = [1,1,1,2,2,3], k = 2))
24
25 #second case
26 #Input: nums = [1], k = 1
27 #Output: [1]
28 ob1: Solution = Solution()
29 print("Leetcode's second case: ", ob1.topKFrequent(nums = [1], k = 1))
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\melan\Desktop\Algorithms\week3 assy> cd "c:/Users/melan/Desktop/Algorithms/week3 assy"
PS C:\Users\melan\Desktop\Algorithms\week3 assy> & C:/Users/melan/AppData/Local/Microsoft/WindowsApps/python3
.10.exe "c:/Users/melan/Desktop/Algorithms/week3 assy/heap.py"
Leetcode's first case: [1, 2]
Leetcode's second case: [1]
PS C:\Users\melan\Desktop\Algorithms\week3 assy>
```

Activate Windows
Go to Settings to activate Windows.

Ln 16, Col 9 Spaces: 4 UTF-8 CRLF Python 3.10.10 64-bit (microsoft store) 5:14 PM 3/22/2023

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))
```