Week-9

Step1: We'll create a frequency table to keep track of the counts. Here's the frequency table for the given input: [1, 1, 1, 2, 2, 3]

Element	Frequency
1	3
2	2
3	1

Step 2: Sort the elements by frequency.

Sorting the elements in the frequency table:

Element	Frequency
1	3
2	2
3	1

Step 3: Take the top k elements, where k = 2 in this case.

The top 2 elements are [1, 2].

Code:

```
import heapq;
from collections import Counter;

def topKFrequent(nums, k):

    counts = Counter(nums)


    heap = []
    for num, count in counts.items():
        heapq.heappush(heap, (count, num))
        if len(heap) > k:
            heapq.heappop(heap)


    top_k = []
    while heap:
        top_k.append(heapq.heappop(heap)[1])
```

```python
    return top_k[::-1]

nums = [1, 1, 1, 2, 2, 3]
k = 2
output = topKFrequent(nums, k)
print(output)
```