Top k frequent element - using sorting

```
class Solution:
    def top_k_frequent_elem(self, nums: List[int], k: int) -> List[int]:

        count = {}

        for num in nums:
            count[num] = 1 + count.get(num, 0)

        arr = []
        for num, cnt in count.items():
            arr.append([cnt, num])
        arr.sort()

        result = []
        while len(result) < K:
            result.append( arr.pop()[1])
        return result

print(Solution(). top_k_frequent_elem(nums=[1,2,2,3,3,3], k=2))
```

Step 1:

self -> solution instance

nums -> [1,2,2,3,3,3]

k -> 2

count -> {} - empty dictionary

Step 2: for num in nums: count[num] = 1 + count.get(num, 0)

- 1st iteration:

num -> 1

count[1] = 1 + count.get(1, 0)

count = {1:1}

- 2nd iteration:

num -> 2

count[2] = 1 + count.get(2, 0)

count = {1:1, 2:1}

- 3rd iteration:

num -> 2

count[2] = 1 + count.get(2, 0)

count = {1:1, 2:2}

- 4th iteration:

num -> 3

count[3] = 1 + count.get(3,0)

count = {1:1, 2:2, 3:1}

- 5th and 6th iteration;

num -> 3

count[3] = 1 + count.get(3,0)

count = {1:1, 2:2, 3:3}

Step 6:

arr -> [] - empty list

INFO: items() method returns a view object (key-value pairs of the dictionary)

Step 7: for num, cnt in count.items():

arr.append([cnt, num])

arr.sort()

- 1st iteration

num -> 1, cnt -> 1

arr -> [[1,1]]

- 2nd iteration

num -> 2, cnt -> 2

arr -> [[1,1], [2,2]]

- 3rd iteration

num -> 3, cnt -> 3

arr -> [[1,1], [2,2], [3,3]]

Info: pop() method removes the element at the specified position.

Step 8:

result -> [] - empty list

Step 9: while len(result) < K:

result.append(arr.pop()[1])

<- last list -> index 1 (num)

1st Iteration:

result -> [3]

2nd iteration

result -> [3,2] ✓