Top k frequent element - bucket sort

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

        count = {}
        freq = [[] for i in range(len(nums) + 1)]

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

        result = []
        for i in range(len(freq)-1, 0, -1):
            for num in freq[i]:
                result.append(num)
                if len(result) == k:
                    return result

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

Step 1:

self → solution instance

nums → [1,1,1,2,3,3]

k → 2

count → {} - empty dict

Step 2: freq = [[] for i in range (len(nums) +1)]

i → increases by 1 for each iteration

freq → [[],[],[],[],[],[],[]]

Step 3: for num in nums:

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

nums

num through each iteration becomes 1, then 1, then 1, then 2 ....

count → {1:3, 2:1, 3:2}

count will basically increase for each repetitive number, the value.

Step 4: for num, cnt in count.items():

freq[cnt].append(num)

we got our empty list of ~~lists~~:

freq = [[],[],[],[],[],[],[]]

1st iteration:

num -> 1 } -> freq[3] add(1)
cnt -> 3 }

2nd iteration:

num -> 2 } freq[1] add(2)
cnt -> 1 }

3rd iteration:

num -> 3 } freq[2] add(3)
cnt -> 2 }

Step 5:

result -> [] - empty list

Step 6: for i in range(len(freq)-1, 0, -1):

i -> 6 because now we iterate in decreasing oxay:

" len(freq)-1 is 7 -1 => 6 so

range (len(7)-1, to 0, in decreasing oder (-1)):

at index 6 we got nothing in our list,

so this will iterate untill i becomes 3

i -> 3

once i is 3, then num becomes 1
but len of result it's still not equal to 2.

so we iterate again

i -> 2, num becomes 3

we add this to our result using

for num in flag[i]:
    result.append(num)

result -> [1, 3] ✓