Top k elements - using heap

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

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

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

        result =[]
        for i in range(k):
            result. append (heapq.heappop(heap)[1])
        return result.

print(Solution(). top_k_freq_elem(self, nums=
```

Info: Heapq

- heapq.heappush (heap, item) - push item onto heap

for example:

heap = [ ]

heapq.heappush (heap, 10)

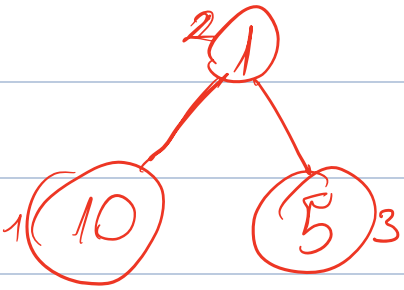
heap → [10]

heapq.heappush (heap, 1)

heap → [1, 10]

heapq.heappush (heap, 5)

heap → [1, 10, 5]



- heapq.heappop (heap) - will return the smallest value and also it will delete that from heap.

for example:

heapq.heappop (heap)

it will remove 1

heap → [5, 10]

we use again heapq.heappop (heap)

it will remove 5

heap → [10]

Step 1:

self -> solution instance

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

K -> 2

count -> {} - empty dict

Step 2: for num in nums:

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

this will iterate 6 times as the length of the list "nums".

num -> 1 then count -> {1:1}

num -> 1 again then count -> {1:2}

num -> 1 again then count -> {1:3}

num -> 2, then count -> {1:3, 2:1}

num -> 3 then count -> {1:3, 2:1}

num -> 3 again then count -> {1:3, 2:1, 3:2}

Step 3:

heap -> [] - empty list

Step 4: for num in count.keys():

heapq.heappush(heap, (count[num], num))

this 2 line of code will iterate for 3 times:

num -> 1 then heap -> [(3,1)]

num -> 2 then heap -> [(1,2), (3,1)]

num -> 3 then heap -> [(1,2), (3,1), (2,3)]

Step 5: if len(heap) > k: ✓ true as len(heap) = 3 > k = 2

Step 6: heapq.heappop(heap) - it will remove (1,2) the smallest

heap -> [(2,3), (3,1)]

Step 7:

result -> [] empty list

Step 8: for i in range(2):

result.append(heapq.heappop(heap)[1])

1st. i -> 0 then result -> [3]

heap will remove index 1 from first tuple (2,3)

i -> 1 then result -> [3,1] ✓