HEAP

1. **TOP K frequent Elements**

   a). #BUCKET SORT : TC: O(N), SC: O(N)

   Step 1: Make count list of empty lists of len(nums) + 1

   Step 2: Make frequency dictionary of each element of nums

   Step 3: Append each frequency v at i of count.

   Step 4: Iterate from last to 0 and append each val in new answer list till len(answer) is equal to k

   b) #Using Heap : TC: O(NlogK), SC: O(N)

   Step1: Make frequency dictionary of each number in nums

   Step2: Make an empty list answer and append (value, freq) as a tuple.

   Step3: Make a empty heap

   Step4: Traverse the answer and if heap length is not equal to k then push the current tuple in the heap

   Step5: If it is equal to length of k then push the current tuple and pop the heap and move to next iteration

   Step6: Make a empty list result

   Step7: Now, till heap is empty pop the tuple 2nd element and append it in result

2. **Merge K sorted Lists**

   a). Using heap: TC: O(NlogK), SC: O(k) where k is total number of lists and N is total number of elements in a list

   Step1: Create empty list

   Step2: Enumerate lists as i , v and push a tuple as (v.val, i) in the heap list

   Step3: Create a dummy node and attach head to it

   Step4: While heap is not empty, pop the element from the heap and assign dummy.next to its value

   Step5: Check if lists[i].next exists or not. If yes, push the tuple as (lists[i].next.val, i) and assign current lists[i] to its lists[i].next node.

Step6: Assign dummy to its next node as dummy.next
Step7: Return head.next

3. **Merge K sorted arrays**

a. <u>Using heap</u>: TC: O(Nlog(K) where k is total number of lists and N is total number of elements in the lists
Step1: Create empty list
Step2: Enumerate nums as i, v. and if len(v) exists then push a tuple as (v.val, i, 0) in the heap list
Step3: Create a result empty list.
Step4. while heap is not empty, pop the element from the heap as v, i, col and append v to the answer
Step5. Check if nums[i][col] is not equal to nums[i][-1] if it’s not equal then push the tuple as (nums[i][col+1], i, col+1]
Step6. At the end return the result list.