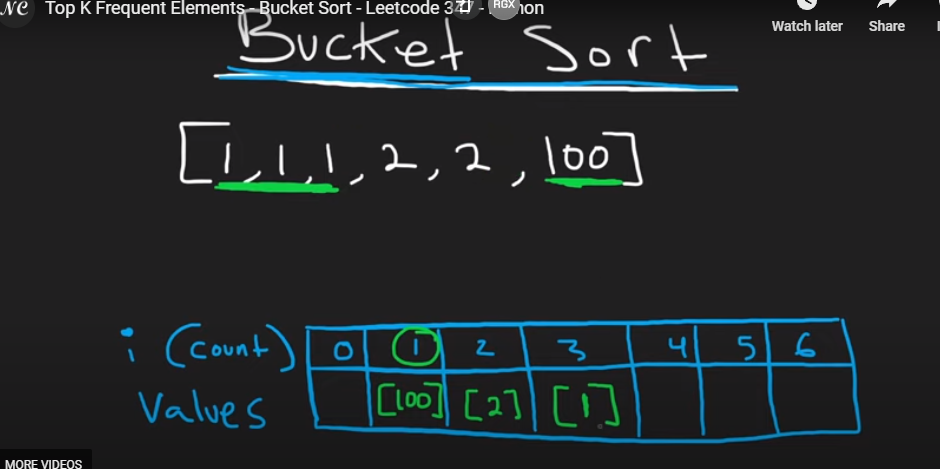
1)we can use a heap to solve it (k log n)

2)use bucket sort :

* bucket sort would have an array with index from 0 to N(length input array)
* hashmap to use no of accourance

and then each element will be stored at the index which is how many times it was repeated

example : if 1 is reaped 3 times 1 will be stored at the index 3



Now we find the top k values we aitrate from 6 in reverse and add the results to our result array

CODE :

Freq = [[] for I in range (len(nums) +1 )] // this generates an empty array of array that has 6 empty arrays in it

Count[n] = 1 + count.get(n,0 ) // this will set the value of nth element to   
def topKFrequent2(self, nums, k):

        """

        :type nums: List[int]

        :type k: int

        :rtype: List[int]

        """

        count = {}

        freq = [[] for i in range(len(nums) + 1)] #generates list of n(length of input array) empty lists

        for n in nums:

            count[n] = 1 + count.get(n, 0) #if n is present in the hash then increment by 1 else default 0

        for n,c in count.items():

            # n appears c no of times

            freq[c].append(n) #append the value 'n' at the index 'c'th array

        res = []

        # reverse traverse the freq array and append the top elements to result

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

            for n in freq[i]:

                res.append(n)

                if len(res) == k:

                    return res