Topk feaguent doment - using sorting class Solution:

def top_K_frequent_elem(self, nums: list Eint] k: int)-> list [int]; count - & g for num in nums; count [num] = 1+ count get (num, o) Sor num, cnt in count. items ():

over append ([cnt, num])

over sort() tosult = [] while lon(result) < K: result append (arr. pop ()[1]) return result

print (Solution (). +p-K-frequent-clom (nums=[1,2,2,3,3,3), K=2))

Step 1:
sell-7 solution instance
self-7 solution instance nums-> [1,2,3,3,3]
$k \rightarrow 2$
Count -> eg - empty dictionary
Stop 21 for num in nums: court Enum] = 1+ court-get (numpo - 15t Horation:
Num -7 1
count [1] = 17 count. get (1,0)
count [1] = 17 $count. get (1,0)count = 21:13$
-2nd iteration:
Num => 2
count[2] = 1+ count.get(2,0) $count = {1:1, 2:1}$
comt = 91:1, 2:19
-3701 Headin:
num = 2
count[2] = 1+ count. get (2,0)
count[2] = 1+ count.get(2,0) $count = {1:1, 2:2}$
<u> </u>

- 4th iteration:

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

count = 21:1, 2:2, 3:13
   -5th and 6th iteration;
         nem -> 3
          Count [3] = 17 count get (3,0)

count = 21:1, 2:2, 3:33
   Stop 6:
over = 7 I J - empty list

(NFD, items!) method returns a view object (key value pairs of the dictionary)

Step 7: for num, ont in Count. items();
                    orr-append ([cont, rum])
                  orr sort)
  - 1st iteation
         num -> 1, cnt ->/
        ar -> [[1,1]]
   -2nd iteration
          num - 72, cn + - 72
          or -> [[1,1], [2,2]]
   - 3701 iteration
```

num - > 3

num > 3, cm + ->3
ord -> [[1,1], [2,2], [3,3]]
Info: popl) method removes the element at the specified position.
Step8;
result -> [] = empty (2)
Step 30 while lem (result) < K:
result. append (ort-nop () [1])
Step 31. while Com (result) < K: result. append (arr. pop () [1]) Clot lest 2 sindex (num)
15t Horation:
result -> [3]
2nd itoation
result -> [3,2] V