

In [5]:

```
1 ##### {a:4,g:9,i:6,p:213,c=6}
2 # [4,6,6,9,213]
3 # [a,c,g,i,p]
4 # k=3
5 # li=[]
6 # for item in d.items():
7 # if item[1]==6:
8 #     li.append(item[0])
9 # li=[i,c]
10
11 def kLargestFrequency(s,k):
12     #Construct the frequency dic
13     unique=[]
14     freq={}
15     for i in s:
16         if i not in unique:
17             freq[i]=s.count(i)
18     values=sorted(freq.values(),reverse=True)
19     uniquevalues=list(set(values))
20     uniquevalues=sorted(uniquevalues,reverse=True)
21     if len(uniquevalues)>=k:
22         kvalue=uniquevalues[k-1]
23     else:
24         return -1
25     for item in freq.items():
26         if item[1]==kvalue:
27             li.append
28     return min(li)
29 print(kLargestFrequency([1,2,3,4,2,1,3],4))
```

-1

In [7]:

```
1  # Function to find second largest number
2
3  def secondLargest(li):
4      #Convert the list into a unique list
5      unique=[]
6      for n in li:
7          if n not in unique:
8              unique.append(n)
9      unique=sorted(unique,reverse=True)
10     return unique[1]
11 secondLargest([1,2,3,4,5,8,7,1,2,3])
12
13 def fifthLeast(li):
14     unique=[]
15     for n in li:
16         if n not in unique:
17             unique.append(n)
18     unique=sorted(unique,reverse=True)
19     return unique[4]
20 fifthLeast([1,2,3,4,5,8,7,1,2,3])
21
22 def secondLargest(li):
23     unique=[]
24     for n in li:
25         if n not in unique:
26             unique.append(n)
27     unique=sorted(unique,reverse=True)
28     if len(unique)>=1:
29         return unique[1]
30     return -1
31 secondLargest([1,2])
32
33 def kLargest(li):
34     unique=[]
35     k=int(input())
36     for n in li:
37         if n not in unique:
38             unique.append(n)
39     unique=sorted(unique,reverse=True)
40     if len(unique)>=k:
41         return unique[k-1]
42     return -1
43 kLargest([1,2,3,4,12,4,5,7,5])
```

3

Out[7]: 5

```
In [8]: 1 def kSmallest(li):
2         k=int(input())
3         unique=[]
4         for n in li:
5             if n not in unique:
6                 unique.append(n)
7         unique=sorted(unique)
8         if len(unique)>=k:
9             return unique[k-1]
10        return -1
11        kSmallest([3,3,9])
```

2

Out[8]: 9

```
In [9]: 1 # Function to identify the element with highest frequency in a list
2 # If many elements have the highest frequency returns the smallest
3 # Highest frequency of following elements([1,2,3,9,8,7,3,4,2,1])->1
4
5 def highestFrequencyElement(li):
6     unique={}
7     for i in li:
8         if i in unique:
9             unique[i]+=1
10        else:
11            unique[i]=1
12        #unique={1:2,2:2,3:2,9:1,8:1,7:,4:1}
13
14        #Getting all frequencies into list
15        freq=unique.values()
16        maxfreq=max(freq)
17
18        #Extract all the keys with maxfreq in a list
19        maxfreqkeys=[]
20
21        #Identify the smallest number
22        for item in unique.items():
23            if item[1]==maxfreq:
24                maxfreqkeys.append(item[0])
25
26        # Select the minimum from the keys with values
27
28        return min(maxfreqkeys) #minfreqE[k-1]
29
30    print(highestFrequencyElement([1,2,3,9,7,3,4,2,1]))
```

1

```
In [10]: 1 # Function to identify the element with second highest frequency
2 # if there are many such elements, return smallest number
3 # [1,2,3,2,1,4,4]-->1
4
5 def secondHighest(li):
6     unique={}
7     for n in li:
8         if n in unique:
9             unique[n]+=1
10        else:
11            unique[n]=1
12    print(unique)
13
14    freq=unique.values()
15    uniquemax=[]
16    for i in freq:
17        if i not in uniquemax:
18            uniquemax.append(i)
19    uniquemax=sorted(uniquemax,reverse=True)
20
21
22    secondmax=uniquemax[1]
23
24    secondmaxkeys=[]
25
26    for item in unique.items():
27        if item[1]==secondmax:
28            secondmaxkeys.append(item[0])
29    return min(secondmaxkeys)
30    print(secondHighest([1,2,3,2,1,4,4,4]))
```

```
{1: 2, 2: 2, 3: 1, 4: 3}
1
```

```

In [2]: 1 # Function to identify Kth highest frequency element in the list
2 # If there are many such elements, return the smallest number
3 #[9,8,7,6,5,2,3,4,9,6,7,7,7,6,7,6]-> k=4->2
4
5
6 def kHighestfreq(li,k):
7     unique={}
8     for n in li:
9         if n in unique:
10             unique[n]+=1
11         else:
12             unique[n]=1
13
14     freq=unique.values()
15     uniquemax=[]
16     for i in freq:
17         if i not in uniquemax:
18             uniquemax.append(i)
19     uniquemax=sorted(uniquemax,reverse=True)
20
21
22     if len(uniquemax)>=k:
23         kmax=uniquemax[k-1]
24         kmaxkeys=[]
25         for item in unique.items():
26             if item[1]==kmax:
27                 kmaxkeys.append(item[0])
28         return min(kmaxkeys)
29     return -1
30 filename='DataFiles/largest.txt'
31 with open(filename,'r') as f:
32     t=int(f.readline())
33     for i in range(t):
34         li=f.readline()
35         k=int(f.readline())
36         print(kHighestfreq(li,k))
37

```

```

s
g
h
e
w
r
n
k
-1

```

```

In [13]: 1 def kLowestfreq(li):
          2     k=int(input())
          3     unique={}
          4     for n in li:
          5         if n in unique:
          6             unique[n]+=1
          7         else:
          8             unique[n]=1
          9     print(unique)
         10
         11     freq=unique.values()
         12     uniquelowest=[]
         13     for i in freq:
         14         if i not in uniquelowest:
         15             uniquelowest.append(i)
         16     uniquelowest=sorted(uniquelowest)
         17     print(uniquelowest)
         18
         19
         20     if len(uniquelowest)>=k:
         21         klowest=uniquelowest[k-1]
         22         klowestkeys=[]
         23         for item in unique.items():
         24             if item[1]==klowest:
         25                 klowestkeys.append(item[0])
         26         return min(klowestkeys)
         27     return -1
         28
         29     print(kLowestfreq([9,8,7,6,5,2,3,4,9,6,7,7,7,6,7,6]))

```

```

4
{9: 2, 8: 1, 7: 5, 6: 4, 5: 1, 2: 1, 3: 1, 4: 1}
[1, 2, 4, 5]
7

```

```

In [ ]: 1 #

```