Day Objectives:-

26th June 2019

- Function to find the Number having the kth largest frequency
- Largest nuber in a list
- Second Largest number in a list
- Kth largest number in a list
- Element with highest frequency
- Second Highest frequency
- Kth Highest frequency¶

Function to find the Number having the kth largest frequency

Largest nuber in a list

Second Largest number in a list

Kth largest number in a list

In [1]:
```
# Function to find the largest Frequency of the element
def largestFrequency(N,K):
    # build the frequency dictionary for all unique characters
    unique=[]
    freq={}
    for i in N:
        if i not in freq.keys() :
            freq[i]=N.count(i)
    # Extract unique frequencies in descending
    values=sorted(freq.values(),reverse=True)
    uniquevalues=list(set(values))
    uniquevalues=sorted(uniquevalues,reverse=True)
    # Identify the kth largest frequency
    if K<=len(uniquevalues):
        kvalue=uniquevalues[K-1]
    else:
        return -1
    # Get all elements with kth largest frequecy
    li=[]
    for item in  freq.items():
        if item[1]==kvalue:
            li.append(item[0])
    # Minimum of Kth Largest frequency
    return min(li)
largestFrequency('aabcdcc',3)
'''with open('Data\input.txt','r') as f:
    t=int(input())
    for i in range(t):
        N=f.readline()
        K=int(input())
        print(largestFrequency(N,K))
            '''
```

Out[1]:
```
'b'
```

In [ ]:
```

```

In [6]:
```
# Function to find the second largest number in the string

def secondLargest(li):
    #Convert the list into a unique list
    unique=[]
    for n in li:
        if n not in unique:
            unique.append(n)
    unique = sorted(unique,reverse=True)
    if len(unique)>1:
        return unique[1]
li=input()
secondLargest(li)


# def KSmallest(li)
```

```
45 23 45 12 33
```

Out[6]:
```
'4'
```

In [ ]:
```

```

In [15]:
```
# def fifthLeast(li)

def fifthLeast(li):
    unique=[]
    for n in li:
        if n not in unique:
            unique.append(n)
    unique = sorted(unique,reverse=True)
    if len(unique)>1:
        return unique[-5]
    return -1
li=[1,2,3,3,5,7,8,2,3,5,7,8]
fifthLeast(li)
```

Out[15]:
```
7
```

In [ ]:
```

```

In [20]:
```
def Klargest(li):
    unique=[]
    for n in li:
        if n not in unique:
            unique.append(n)
    unique = sorted(unique,reverse=True)
    if len(unique)>K:
        return unique[K-1]
    return -1
li=input().split()
K=int(input())
Klargest(li)
```

```
2 3 4 5 6 7 8 1 2 3 9
3
```

Out[20]: '7'

In [ ]:

In [23]:
```
# Function for  the Kth largest element in the list
def Ksmallest(li):
    # Extract unique elements in the list
    unique=[]
    for n in li:
        if n not in unique:
            unique.append(n)
    # Sort the Unique list in descending order (highest to lowest)
    unique = sorted(unique) # no need to reverse this because smallest  element i
    # Check if length of unique list is greater than K
    if len(unique)>=K:

        return unique[K-1]
    return -1
li=input().split()
K=int(input())
Ksmallest(li)
```

```
1
```

Out[23]: -1

In [ ]:

Element with highest frequency

Second Highest frequency

Kth Highest frequency

In [ ]:

In [ ]:
```
# Function to identify the element with highest frequency
# If any have the highest fequency,return the smallest
# highestFrequency([1,2,3,9,8,7,3,4,2,1])


def highestFrequency(li):
    # Extracting the
    unique={}
    #freq={}
    for n in li:
        if n in unique.keys():
            unique[n]+=1
        else:
            unique[n]=1
# Getting all frequencies into a
    freq = unique.values()
    maxfreq=max(freq)
    maxfreqE = []

# Identify keys with  maximum frequency
    for item in unique.items():
        if item[1] == maxfreq:
            maxfreqE.append(item[0])

# Select the minimum from the keys with maximum frequency
    return min(maxfreqE)# Instead minfreqE [K-1]
    #return max(maxfreqE)
    #maxfreqE=sorted(maxfreqE,reverse=True)
    return maxfreqE[K-1]
#K=int(input())
li=[1,2,3,4,5,7,3,2,4,5,1,2,5,3,4,8,2,1,3,4,5,6,5]
highestFrequency(li)
```

In [55]:
```
# Function to identify the element with highest frequency
# If any have the highest fequency,return the smallest
# highestFrequency([1,2,3,9,8,7,3,4,2,1])


def highestFrequency(li):
    unique=[]
    fre=[]
    frecount=[]
    for i in li:
        if i not in unique:
            unique.append(i)
    unique=sorted(unique)
    print(unique)
    for i in unique:
        fre=li.count(i)
        print(i,'=',fre,end="\n")
li=[1 ,9,9,3,4,5,2,4,5,2,6,1,3,]
highestFrequency(li)
```

```
[1, 2, 3, 4, 5, 6, 9]
1 = 2
2 = 2
3 = 2
4 = 2
5 = 2
6 = 1
9 = 2
```

In [140]:
```
# Function to identify the element with highest frequency
# If any have the highest fequency,return the smallest
# highestFrequency([1,2,3,9,8,7,3,4,2,1])


def highestFrequency(li):
    # Extracting the
    unique={}
    for n in li:
        if n  in unique:
            unique[n]+=1
        else:
            unique[n]=1
    print(unique)
    maxfreq=max(unique.values())
    # Extract all the keys with maximum frequency
    maxfreqkeys=[]
    for item in unique.items():
        if item[1]==maxfreq:
            maxfreqkeys.append(item[0])
    return min(maxfreqkeys)

li=input().split()
K=int(input())
highestFrequency(li)
```

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

Out[140]:
```
'1'
```

In [143]:
```
# Function to identify second highest frequency element.If there are many such e

# Function to identify the element with highest frequency
# If any have the highest fequency,return the smallest
# highestFrequency([1,2,3,9,8,7,3,4,2,1])


def SecondFrequency(li):
    # Extracting the
    unique={}
    u={}
    for n in li:
        if n  in unique:
            unique[n]+=1
        else:
            unique[n]=1
    print(unique)

    u=sorted(unique.values())
    print(u)
    usecond=[]
    for i in u:
        if i not in usecond:
            usecond.append(i)
    print(usecond)
    us=usecond[-2]
    print(us)
    # Extract all the keys with maximum frequency
    maxfreqkeys=[]
    for item in unique.items():
        if item[1]==us:
            maxfreqkeys.append(item[0])
    return min(maxfreqkeys)
li=[1,1,2,2,3,4,4,9]
SecondFrequency(li)
```

```
{1: 2, 2: 2, 3: 1, 4: 2, 9: 1}
```

Out[143]:
```
{1: 2, 2: 2, 3: 1, 4: 2, 9: 1}
```

In [154]:
```
# Function to identify second highest frequency element.If there are many such e

# Function to identify the element with highest frequency
# If any have the highest fequency,return the smallest
# highestFrequency([1,2,3,9,8,7,3,4,2,1])


def KthLargest(li):
    # Extracting the
    unique={}
    u=[]
    for n in li:
        if n  not in unique:
            unique[n]=1
        else:
            unique[n]+=1
    print(unique)
    # Extract all frequencies in a unique frequencies in a list
    u=sorted(unique.values(),reverse=True)
    print(u)
    usecond=[]
    for i in u:
        if i not in usecond:
            usecond.append(i)
    print(usecond)
    if len(usecond)>=K:
        us=usecond[K]
        print(us)
        # Extract all the keys with maximum frequency
        maxfreqkeys=[]
        for item in unique.items():
            if item[1]==us:
                maxfreqkeys.append(item[0])
        return min(maxfreqkeys)
    else:
        return -1
K=int(input())
li=input().split()
KthLargest(li)
```

```
3
1 1 2 2 3 4 4 9 9
{'1': 2, '2': 2, '3': 1, '4': 2, '9': 2}
[2, 2, 2, 2, 1]
[2, 1]
```

Out[154]:
```
-1
```

In [162]:
```
# Function to identify second highest frequency element.If there are many such e

# Function to identify the element with highest frequency
# If any have the highest fequency,return the smallest
# highestFrequency([1,2,3,9,8,7,3,4,2,1])


def HackerearthProblem(li):
    # Extracting the
    unique={}
    u={}
    for n in li:
        if n  in unique:
            unique[n]+=1
        else:
            unique[n]=1
    #print(unique)
    u=sorted(unique.values())
    #print(u)
    usecond=[]
    for i in u:
        if i not in usecond:
            usecond.append(i)
    #print(usecond)
    if len(usecond)>K:
        us=usecond[-K]
        #print(us)
        # Extract all the keys with maximum frequency
        maxfreqkeys=[]
        for item in unique.items():
            if item[1]==us:
                maxfreqkeys.append(item[0])
        return min(maxfreqkeys)
    else:
        return -1


with open('Data/i.txt','r') as f:
    T=int(f.readline())
    for i in range(T):
        li = f.readline()
        K=int(f.readline())
        print(HackerearthProblem(li))
```

```
s
g
h
e
w
r
n
k
-1
-1
```

In [ ]: