Searching

1] Binary Search In Python

```
def bSearch(l,x):
    low=0
    high=len(l)-1
    while low<=high:
        mid=(low+high)//2

        if l[mid]==x:
            return mid
        elif(l[mid]<x):
            low=mid+1
        else:
            high=mid-1
    return -1


l=[10,20,30,40,50,60]

print(30,bSearch(l,30))
print(20,bSearch(l,20))
print(10,bSearch(l,10))
print(60,bSearch(l,60))
print(40,bSearch(l,40))
print(55,bSearch(l,55))
print(-50,bSearch(l,-50))
```

OUTPUT :

```
30 2

20 1

10 0

60 5

40 3
```

55 -1

-50 -1

2] Recursive Binary Search :

```
def bSearch(l,x,low,high):
    if low>high:
        return -1

    mid=(low+high)//2
    if l[mid]==x:
        return mid

    elif l[mid]>x:
        return bSearch(l,x,low,mid-1)

    else:
        return bSearch(l,x,mid+1,high)

def bSearchMain(l,x):
    return bSearch(l,x,0,len(l)-1)

l=[10,20,30,40,50,60]

print(30,bSearchMain(l,30))
print(20,bSearchMain(l,20))
print(10,bSearchMain(l,10))
print(60,bSearchMain(l,60))
print(40, bSearchMain(l, 40))
print(55, bSearchMain(l, 55))
print(-50, bSearchMain(l, -50))
```

OUTPUT :

30 2

20 1

10 0

60 5

```
40 3

55 -1

-50 -1
```

3] Index Of First Occurance :

```
def firstIndex(l,x):
    for i in range(len(l)):
        if l[i]==x:
            return i

    return -1

l=[10,10,20,20,20,30,30,30,30,]
print(20,firstIndex(l,20))
print(30,firstIndex(l,30))
print(10,firstIndex(l,10))
```

OUTPUT :

```
20 2

30 5

10 0
```

4] Index Of First occurance Efficient :

```
def firstIndex(l,x):
    low=0

    high=len(l)-1
    while low<=high:

        mid=(low+high)//2

        if l[mid]>x:
            high=mid-1
        elif l[mid]<x:
            low=mid+1

        else:
            if mid==0 or l[mid-1]!=l[mid]:
                return mid
            else:
                high=mid-1
    return -1

l=[5,10,10,20,20]
print(10,firstIndex(l,10))
print(20,firstIndex(l,20))
print(50,firstIndex(l,50))
```

OUTPUT :

```
10 1

20 3

50 -1
```

5] Index Of Last Occurance :

```
def lastOccur(l,x):

    for i in reversed(range(len(l))):  #Python reversed() method returns an iterator that accesses the given sequence in the reverse order.
        if l[i]==x:
            return i
    return -1

l=[10,15,20,20,40,40]

print(20,lastOccur(l,20))
print(40,lastOccur(l,40))
```

OUTPUT :

20 3

40 5

6] Index Of Last Occurance Efficient :

```
def lastOccur(l,x):
    low=0
    high=len(l)-1

    while low<=high:
        mid=(low+high)//2

        if l[mid]<x:
            low=mid+1

        elif l[mid]>x:
            high=mid-1

        else:
            if mid==len(l)-1 or l[mid+1]!=l[mid]:
                return mid
            else:
                low=mid+1

    return -1

l=[5,10,10,10,10,20,20]

print(10,lastOccur(l,10))
print(20,lastOccur(l,20))
print(25,lastOccur(l,25))
```

OUTPUT :

```
10 4

20 6

25 -1
```

7] Count the number of Occurance :

```
def countOccur(l,x):
    cnt=0

    for e in l:
        if e==x:
            cnt+=1

    return cnt

l=[10,20,20,20,30,30]

print(10,countOccur(l,10))
print(20,countOccur(l,20))
print(30,countOccur(l,30))
print(25,countOccur(l,25))
```

OUTPUT :

```
10 1

20 3

30 2

25 0
```

8] count occurance in sorted array of efficient :

```
def firstIndex(l,x):
    low=0

    high=len(l)-1

    while low<=high:

        mid=(low+high)//2

        if l[mid]>x:
            high=mid-1

        elif l[mid]<x:
            low=mid+1

        else:
            if mid==0 or l[mid-1]!=l[mid]:
                return mid

            else:
                high=mid-1
    return -1

def lastOccur(l,x):

    low=0
    high=len(l)-1

    while low<=high:
        mid=(low+high) //2

        if l[mid]<x:
            low=mid+1
        elif l[mid]>x:
            high=mid-1

        else:
            if mid==len(l)-1 or l[mid]!=l[mid + 1]:
                return  mid

            else:
                low=mid + 1

    return -1
```

```
def countOccurr(l,x):
    first=firstIndex(l,x)

    if first==-1:
        return 0

    else:
        return lastOccur(l,x) -first+1


l=[10,20,20,20,30,30]

print(10,countOccurr(l,10))
print(20,countOccurr(l,20))
print(30,countOccurr(l,30))
print(25,countOccurr(l,25))
```

OUTPUT :

```
10 1

20 3

30 2

25 0
```