

Arrays

In [34]:

```
#Binary Search
arr =[20,30,40,50,60,70]
def binary_seach(arr,left,right):
    x=50
    left = 0
    right=len(arr)
    mid = right//2
    while left<=right:
        if (arr[mid]==x):
            return mid
        elif (arr[mid]<x):
            left = mid+1
        else:
            right = mid-1
    print(binary_seach(arr,0,5))
```

3

In [35]:

```
#Binary Search with recursion
def binary_re(arr1,x,i,j):

    while i<=j:
        mid = i + (j-i)//2
        if arr[mid]== x:

            return mid

        elif(arr[mid]<x):
            return binary_re(arr1,x,mid+1,j)

        else:
            return binary_re(arr1,x,i,mid -1)

arr1 =[20,30,40,50,60,70,80]
i=0
j= len(arr1) -1
x=50
result = binary_re(arr1,x,i,j)
print(result)
```

3

In [36]:

```
def modify_b(arr,left,right):
    while left<right:
        mid = left + (right-left)//2

        if(arr[mid]!="i"):
            return modify_b(arr,mid +1 , right)
        else:
            if(arr[mid-1]!="i"):
                return mid
            else:
                return modify_b(arr,left,mid -1)
```

```
arr2 = [-20,30,1,2,7,0,9,"i","i","i"]
left=0
right = len(arr2) - 1
result = modify_b(arr2,left,right)
print(result)
```

7

Exponential Binary Search

In [37]:

```
def exponential_search(arr, i , j,x):

    for item in arr(range(0::i**2)):
        mid = i +(j-i)//2

        if arr[mid]==x:

            if arr[mid - 1]!=x:
                return mid
            else:
                return exponential_search
```

Cell In[37], line 3

```
for item in arr(range(0::i**2)):
```

SyntaxError: invalid syntax

In []:

#Two_Pointer

```
def two_pointer(arr,l,r):

    while l<=r:
        if arr[l]+arr[r]==sum_value:
            return arr[l],arr[r]

        elif (arr[l]+arr[r]<sum_value):
            return two_pointer(arr,l+1,r)

        else:
            return two_pointer(arr,l,r-1)

    return -1
```

```
arr3 =[20,40,60,80,90,120]
sum_value=90
l = 0
r = len(arr3) - 1
result = two_pointer(arr3,l,r)
print(result)
```

-1

In []:

```
def first_element(nums : list[int],target = int):
```

In []:

In []:

```
num1 = [2,3,5,4,2]
num2 =[2,2]
k=[]

for s in num2:

    if s in num1:
        k.append(s)

j = set(k)
print(list(j))
```

[2]

BUY and SELL stock

In [2]:

```
stock =[1,5,3,6,4,8]
k =[]
l = 0
r = 1

while r<(len(stock)):

    if stock[l]< stock[r]:

        k.append(stock[r]- stock[l])

    elif stock[l]> stock[r]:
        l = r

    r +=1
print(max(k))
```

7

In []:

```
#more optimize way to solve
```

Tenary Search

In [6]:

```
def tenarySearch(arr,l,r,key):

    while l<=r :

        mid1 = l + (r-l)//3
        mid2 = r - (r-l)//3

        if(arr[mid1]==key):
            return mid1

        elif(arr[mid2]==key):
```

```

        return mid2

    elif(key<arr[mid1]):
        return tenarySearch(arr,l,mid1-1,key)

    elif(key>arr[mid1]):
        return tenarySearch(arr,mid1+1,r,key)

    elif(key<arr[mid2]):
        return tenarySearch(arr,mid1+1,mid2-1,key)

    elif(key>arr[mid2]):
        return tenarySearch(arr,mid2+1,r,key)

    return -1

arr =[20,22,24,36,48,56,65,69,88,99]
key = 69
l=0
r=len(arr)-1
result = tenarySearch(arr,l,r,key)
print(result)

```

7

In []:

2D Array

In [39]:

```

def findTarget(matrix,target):
    row= len(matrix)
    column=len(matrix[0])

    i=0
    j = (row*column)-1

    while i<=j:

        mid = i+ (j-i)//2

        midRow = mid//column
        midColumn = mid % column
        midvalue = matrix[midRow][midColumn]

        if(midvalue==target):

            return "true"

        elif(midvalue>target):

            j=mid-1

        else:

            i= mid+1

    return "false"

matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]]

```

```
target = 23  
result=findTarget(matrix,target)  
print(result)
```

true