**DAY-11:**

**1.two-pointer problem**.

def two\_pointer(arr,target):

arr.sort()

i,j = 0, len(arr)-1

while i<j:

if arr[i]+arr[j] == target: return True

if arr[i]+arr[j] < target: i+=1

if arr[i]+arr[j] > target: j-=1

return False

**2.Sliding Window.**

def sliding\_window(arr,k):

\_sum = 0

ps = 0

i,j = 0,k-1

while j<len(arr):

if i==0:

\_sum = sum(arr[i:j+1])

ps = \_sum

else:

cs = ps - arr[i-1]+arr[j]

\_sum = max(\_sum,cs)

ps = cs

i+=1

j+=1

return \_sum

print(sliding\_window(

[-3,20,3,-9,18,-45,23,67],3

))

**3.QUERY SUBARRAY.**

def query\_subarray\_sum(arr,queries):

n = len(arr)

ps = [0 for i in range(n)]

for i in range(n):

if i == 0:

ps[i] = arr[i]

else:

ps[i] = ps[i-1]+arr[i]

for query in queries:

i = query[0]

j = query[1]

if i ==0:

print(ps[j],end=" ")

else:

print(ps[j]-ps[i-1],end=" ")

# arr -> [-1,2,32,4,5,5,4,3,2]

''' queires=[

[0,4],

[2,5],

[3,6]

]

'''

query\_subarray\_sum([

2,4,5,1,6,3,7,8

],

[

[0,4],[1,3],[4,6]

]

)

**4.missing Number.**

def brute\_force(n,arr):

for i in range(1,n+1):

if i not in arr:

return i

def xor\_approach(n,arr):

ans = 0

for i in range(1,n+1):

ans = ans^i

if i!=n:

ans = ans^arr[i-1]

return ans

print(xor\_approach(10,[3,4,1,2,6,5,8,10,9]))

**5.Kadaness**

def kadanes(arr):

\_sum = float("-inf")

cs = arr[0]

n = len(arr)

for i in range(1,n):

if cs < 0:

cs = 0

cs = cs+arr[i]

if arr[i] < 0:

\_sum = max(\_sum,cs+arr[i])

return max(\_sum,cs)

print(kadanes([-1,-2,3,-4,-6,18,65,110]))