

## Array: Prefix Sum & Carry

## Forward

- Q) Given an array of  $N$  integers and  $Q$  queries. For each query calculate the sum of elements in the range  $[L, R]$

Note: L & R are indices such that  $L \leq R$ .

$$(1 \leq N, Q \leq 10^5)$$

$$Ans \rightarrow arr[10] \rightarrow [-3 \ 6 \ 2 \ 4 \ 5 \ 2 \ 8 \ -9 \ 3 \ 1]$$

$$Q = [[4, 8], [3, 7], [1, 3], [0, 4], [7, 7]]$$

## Queries :

Given:

## 1) Query 2D List

L	R	Sum
4	8	9
3	7	10
1	3	12
0	4	14
7	7	-9

$$[\lfloor \ell, x \rceil, \lceil \ell, x \rceil, \lceil \ell, x \rceil]$$

2) List of numbers.

Task: Print sum for  $l$  to  $r$  for each query.

## \* Brute force approach -

def print\_sum (li, q) :

# Solve all queries  $\rightarrow$  iterate over q

for i → 0 to len(q)

$$l = q [i] [0]$$

$$g = q^k [i] [1]$$

# beginning of every query

$$\text{sum} = 0$$

# calculate sum for that query [l, r]

for  $j \rightarrow l$  to  $r$ :

$$\text{sum} \pm \text{dif}[j]$$

print sum.

TC: (i) length of query list  $\rightarrow Q$

(ii) length of number list  $\rightarrow N$

$O(Q * N)$

SC:  $O(1)$

$$Q * N \rightarrow 10^5 * 10^5 = 10^{10} > 10^8$$

TLE error

### \* Array - Prefix sum

Overs	1	2	3	4	5	6	7	8	9	10
Runs/Score	2	8	14	29	31	49	65	79	88	97

$$\begin{aligned} \text{Runs scored in 7th over} &= \text{score}[7] - \text{score}[6] \\ &= 65 - 49 = 16 \end{aligned}$$

$$\begin{aligned} \text{Runs scored in 6-10th over} &= \text{score}[10] - \text{score}[5] \\ &= 97 - 31 = 66 \end{aligned}$$

Runs scored in  $i^{\text{th}} - r^{\text{th}}$  over =  $\text{score}[r] - \text{score}[i-1]$

Runs in  $r^{\text{th}}$  over only =  $\text{score}[r] - \text{score}[r-1]$

### \* How to create psum()

$\text{arr}[10] \rightarrow [-3 6 2 4 5 2 8 -9 -3 11]$   
  
 $\text{psum}[10] \rightarrow [-3 3 5 9 14 16 24 15 18 19]$

current value of arr + prev value of psum

if  $i=0$ :  
 $\text{psum}[0] = \text{arr}[0]$

other cases:

$\text{psum}[i] = \text{arr}[i] + \text{psum}[i-1]$

```
def prefixsum(arr):
    n = len(arr)
    psum = [0] * n
    psum[0] = arr[0]
    for i in range(1, n):
        psum[i] = arr[i] + psum[i-1]
    return psum
```

TC:  $O(N)$

## A Optimisation for 1<sup>st</sup> Question -

By creating prefix arrays

$$\text{arr}[10] \rightarrow [-3 \ 6 \ 2 \ 4 \ 5 \ 2 \ 8 \ -9 \ 3 \ 1]$$

$$\text{psum}[10] \rightarrow [-3 \ 3 \ 5 \ 9 \ 14 \ 16 \ 24 \ 15 \ 18 \ 19]$$

0 1 2 3 4 5 6 7 8 9

Queries - 5

L R

$$4 \ 8 \rightarrow \text{psum}[8] - \text{psum}[3] = 18 - 9 = 9$$

$$3 \ 7 \rightarrow \text{psum}[7] - \text{psum}[2] = 15 - 5 = 10$$

$$1 \ 3 \rightarrow \text{psum}[3] - \text{psum}[0] = 9 - (-3) = 12$$

$$0 \ 4 \rightarrow \text{psum}[4] = 14 \quad \text{edge case if } l=D$$

$$7 \ 7 \rightarrow \text{psum}[7] - \text{psum}[6] = 15 - 24 = -9$$

$\rightarrow N \quad \rightarrow Q$   
 $\text{def query\_sum}(\text{arr}, \text{queries}):$

# Step 1: Create prefix sum

$$N = \text{len}(\text{arr})$$

$$\text{psum} = [0] * N$$

$$\text{psum}[0] = \text{arr}[0]$$

$$\text{psum}[i] = \text{arr}[i] + \text{psum}[i-1]$$

# Step 2: Answering queries .

for query in queries:

l, r = query

if l = 0 :

$$\text{sum} = \text{psum}[r]$$

else :

$$\text{sum} = \text{psum}[r] - \text{psum}[l-1]$$

point(sum)

TC :  $O(N+Q)$

SC :  $O(N)$

- Q) Given an arr [N] & Q queries with start (s) & end (e) index. For every day query point sum of all even indexed elements from s to e.

arr [] → 

2	3	1	6	4	5	2
0	1	2	3	4	5	6

Queries :

s + e → sum of even indexed elements (s to e)

1 | 3 → arr[2] = 1

2 | 5 → arr[2] + arr[4] = 1 + 4 = 5

0 | 4 → arr[0] + arr[2] + arr[4] = 2 + 1 + 4 = 7

3 | 3 → no even index ⇒ 0.

- BF idea ⇒ For all queries: # Q

go in range but add only even index # N

TC :  $O(Q * N)$

## \* Optimized Idea-

Whenever you have range sum queries always think of psum.

Create psum of only even indexed elements

arr [ ] →	0	1	2	3	4	5
	2	3	1	6	4	5
	↓	↓	↓			

  

psum [ ] →	0	1	2	3	4	5
	2	2	3 + 3	7 → 7		

$$\text{psum}[0] = \text{arr}[0]$$

if  $i$  is odd     $\text{psum}[i] = \text{psum}[i-1]$

if  $i$  is even     $\text{psum}[i] = \text{psum}[i-1] + \text{arr}[i]$

```
def query_psum(arr, queries):
```

$$N = \text{len}(\text{arr})$$

$$\text{psum} = [0] * N$$

$$\text{psum}[0] = \text{arr}[0]$$

```
for i in range(1, N):
```

```
    if i % 2 == 0:
```

$$\text{psum}[i] = \text{arr}[i] + \text{psum}[i-1]$$

```
    else:
```

$$\text{psum}[i] = \text{psum}[i-1]$$

```
for l, r in queries:
```

```
    if l == 0:
```

$$\text{sum}+ = \text{psum}[r]$$

```
    else:
```

$$\text{sum}+ = \text{psum}[r] - \text{psum}[l-1]$$

```
print(sum+)
```

TC:  $O(N+Q)$

SC:  $O(N)$

Q) Given a character array  $ch[N]$  of size  $N$ , we have to calculate the number of pairs of indices  $i, j$  & &  $ch[i] == 'a'$  & &  $ch[j] == 'g'$

Constraints :  $1 \leq N \leq 10^5$

$'a' \leq ch[i] \leq 'z'$

Observation : 1. Max possible length of array  $\rightarrow 10^5$   
 2. Character array have lowercase alphabets

$ch: T|a|a|g|l|d|c|a|g|l|$

pairs of  $(i, j)$  where  $i < j$ ,  $ch[i] == 'a'$  &  $ch[j] == 'g'$

$\hookrightarrow (1, 3), (6, 7), (2, 3), (1, 7), (2, 7) \Rightarrow 5$  pairs

Idea : if  $ch[i] != 'a'$   $\rightarrow$  skip the iteration

Condition : C1 :  $i < j$   
 C2 :  $ch[i] == 'a'$  &  $ch[j] == 'g'$

arr: [a|d|g|a|g|a|g|f|g|]

Bonute force :

```
def pairag(s):
    N = len(s)
    result = 0
    for i in range(N):
        if s[i] == 'a':
            for j in range(i+1, N):
                if s[j] == 'g':
                    result += 1
    return result
```

TC:  $O(N^2)$  SC:  $O(1)$

Q) Can we do it in TC:  $O(N)$  & SC:  $O(1)$ ?

Yes we can, carry forward technique

Idea: s: [a d g a l g a l g a l g f l g]

count of g: 4 count of pairs: 9

## \* Pseudocode:

```
def countofagpairs(s):
    countofg, countofpairs = 0, 0
    n = len(s)
    (from n-1 to 0)
    for i in range (n-1, -1, -1):
        if s[i] == 'g':
            countofg += 1
        elif s[i] == 'a':
            countofpairs += countofg
    return countpairs.
```