Agenda

1) Preprocessing [concept]
2) Do some questions based on this concept

Q1. Given an array of N element & Q queries
For each of the query, print sum of ele from index s to e [s & e will be given to you for each query]

Eg

	0	1	2	3	4	5	6	7	8	9
A[ ] = {	-3	6	2	4	5	2	8	-9	3	1 }

Q queries
(5)

	0	1	2	3	4	
start[ ] =	4	2	1	0	7	←
end [ ] =	8	7	3	4	7	←

Queries

s	<=	e	sum
4		8	9
2		7	12
1		3	12
0		4	14
7		7	-9

Brute force : For each query, find the sum from s to e

```
for( K = 0 ; K < Q ; K++){    // Q iterations
    s = start[K]
    e = end[K]
    sum = 0
```

```
for ( i = s ; i <= e ; i++ ) {
    sum += A[i]
}
print(sum)
}
```

// [s e]

e - s + 1

0 ... N-1

[0 N-1]

N

TC

O(QXN)

SC O(1)

```
      0  1  2  3  4  5
A[] = {4, 1, 3, -1, 6, 5}
```

Q = 4

start[] = | 0 | 0 | 0 | 0 |

end[] = | 5 | 5 | 5 | 5 |

1 → N

Q → QXN

Cricket ?

Given Indian Cricket team scored for first 10 overs of batting, you've to answer some questions

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

Runs scored in last over. 97 - 88 = 9

[10 10] = score[10] - score[9]

Runs scored in last 5 overs: 97 - 31 = 66

[6 10] = scores[10] - scores[5]

Runs scored from [3 6] : 49 - 8 = 41

[3 6] = scores[6] - scores[2]

Runs scored in 7th : 65 - 49 = 16

[7 7] = scores[7] - scores[6]

score [4 9] = score[9] - scores[3]

88 - 14

Since we're given cummulative score
to find answer for some range
we're just doing subtraction

	0	1	2	3	4	5	6	7	8	9
A[ ] = {	-3	6	2	4	5	2	8	-9	3	1 }
PF[ ] = {	-3	3	5	9	14	16	24	15	18	19 }

prefix sum ⇒

PF[0] = A[0]

PF[1] = A[0] + A[1]
PF[0]

PF[2] = A[0] + A[1] + A[2]
PF[1]

PF[3] = A[0] + A[1] + A[2] + A[3]

PF[2]

⋮

PF[i-1] = A[0] + A[1] + A[2] + A[3] + ... A[i-2] + A[i-1]

PF[i-2]

PF[i] = A[0] + A[1] + ... + A[i-1] + A[i]

PF[i-1]

$$PF[i] = PF[i-1] + A[i]$$

```
int PF[N]
PF[0] = A[0]
for(i=1; i<N; i++){
    PF[i] = PF[i-1] + A[i]
}
```

TC → O(N)

SC → O(N)

	0	1	2	3	4	5	6	7	8	9
A[] =	-3	6	2	4	5	2	8	-9	3	1
PF[] =	-3	3	5	9	14	16	24	15	18	19

Queries

s	<= e	sum
4	8	PF[8] - PF[3] = 18 - 9 = 9
2	7	PF[7] - PF[1] = 15 - 3 = 12

1 3 PF[3] - PF[0] = 9 - (-3) = 12

i j PF[j] - PF[i-1]

0 4 PF[4] - ~~PF[0-1]~~ PF[-1]

PF[4]

sum(i,j) = PF[j] - PF[i-1]

and

if i = 0 then PF[j]

① Create PF[] // N iterations ⇒ O(N)

② for(K = 0 ; K < Q ; K++){ // Q iterations ⇒ O(Q)

```
    s = start[K]
    e = end[K]
    if (s == 0) {
    |   sum = PF[e]
    }
    else {
    |   sum = PF[e] - PF[s-1]
    }
    print (sum)
```

constant

PF[]

TC: $O(N) + O(Q)$ → $O(N+Q)$ SC: $O(N)$

Break!

9:30

=

Application of Prefix sum

Range sum Queries → where we have to tell sum from s to e in Array.

Amazon | Adobe

Q2. Given an array of size N, count no. of equilibrium index.

An index is said to be equilibrium index if -

sum of all ele b4 ith index = sum of all elements after ith index

i

[0 i-1] [i+1 N-1]

	0	1	2	3	4	5
	4	5	1	7	2	9
leftSum	0					
rightSum						0

	0	1	2	3 ✓	4	5	6 ✓
A[]:	-7	1	5	2	-4	3	0
leftSum	0	-7	-6	-1	1	-3	0
rightSum	7	6	1	-1	3	0	0

ans: 2

	0	1	2	3	4	5	6
A[]:	3	-1	2	-1	1	2	1
leftSum	0	3	2	4	3	4	6
right	4	5	3	4	3	1	0

ans: 2

i

leftSum : [0 i-1]

rightSum [i+1 N-1]

sum(0, i-1)

= PF[i-1] - PF[0-1]

sum(i+1, N-1) =

PF[N-1] - PF[i+1-1]

PF[N-1] - PF[i]

sum(s, e) = PF[e] - PF[s-1]

$sum(s,e) = PF[e] - PF[s-1]$

N = 7

	0	1	2	3	4	5	6
A[]	-7	1	5	2	-4	3	0
PF[]	-7	-6	-1	1	-3	0	0
leftsum	0	-7	-6	-1	1	-3	0
rightsum	7	6	1	-1	3	0	0

① construct PF[]

② count = 0

```
for (i=0; i<N; i++) {
    if (i==0) leftsum = 0
    else {leftsum = PF[i-1]}
    rightsum = PF[N-1] - PF[i]

    if ( leftsum == rightsum ) {
        count ++
    }
}
```

Edge case

i = 0

leftsum = PF[i-1]

PF[-1]

0

i = N-1 — No issue Here

rightsum = PF[N-1] - P[i]

PF[N-1] - PF[N-1]

return count = 0

TC: O(N) + O(N) = O(N)

SC: O(N)

sum(s, e) = PF[e] - PF[s-1]

TODO

HW

Do it without using PF array.

Flipkart

Que. Given array of N ele & Q queries, for each query, calc sum of all even indices in given array

0	1	2	3	4	5	6	7
3	4	-2	8	6	2	1	3

A[]:

s	e	sum
2	5	4
3	7	7
0	7	8

odd indices have to role to play.

0	1	2	3	4	5	6	7
3	0	-2	0	6	0	1	0

A[]:

0	1	2	3	4	5	6	7
3	3	1	1	7	7	8	8

PF[]:

s	e	PF[e] - PF[s-1]

2	5	7 − 3 = 4
3	7	8 − 1 = 7
0	7	8

	0	1	2	3	4	5	6	7
A[]:	3	4	-2	8	6	2	1	3
PF_even[]:	3	3	1	1	7	7	8	8

PF_even [ ]

① PF_even[0] = A[0]

```
for( i=1; i<N; i++){
    if (i % 2 == 0) {
        PF[i] = PF[i-1] + A[i]
    }
    else {
        PF[i] = PF[i-1]
    }
}
```

② Answer the Queries.

TC → O(N) + O(Q)

SC → O(N)

TODO

Q. Find sum of odd indexed element from S to C

PF_odd[ ]

?

Order of Terms

N = 32

constant < $\log_2 N$ < $\sqrt{N}$ < N < N log N < $N\sqrt{N}$ < N^2 < 2^N < N! < N^N

$\log_2 32$ $\sqrt{32}$ 32 32x5 32x5.xx $(32)^2$ 2^{32}

5 5.xx 32 160 181 1024 4×10^9 >> 4×10^9

n = 64

```
i = n
while (i > 0) :
    if (i % 2 == 0) :
        for j (1, n^2 + 1, 2) :
    i = i / 2
```

i

n n^2

n/2 n^2

n/4 n^2

n = 15

↓ /2

7

↓ /2

3

↓ /2

1.

n = 64

↓ /2

32

↓ /2

16

↓ /2

8

↓ /2

4

↓ /2

2

↓ /2

1

```
i = 0
while(i * i <= N){
    for(int j = 0; j <= N; j++){
        for(k = 0; k <= N; k++, i++){
            // O(1)
        }
    }
    i++
}
```

TC → $O(N^2)$

ک