# Reference Question:

Given array of size N & Q queries of the format s & e

queries → Question

s → start, e → end

Return the sum of elements from indices s to e [s, e] for each query

Eg

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | −3 | 6 | 2 | 4 | 5 | 2 | 8 | −9 | 3 | 1 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Q=3

| s | e | |
|---|---|---|
| 1 | 3 | sum = 12 |
| 2 | 7 | sum = 12 |
| 0 | 2 | sum = 5 |

```
for (i=0 ; i<Q ; i++) {
    scan (s,e)   // take s&e as input
    int sum=0
    for (j=s ; j<=e ; j++) {
        sum += a[j]
    }
    print sum
}
```

TC: $O(NQ)$

SC: $O(1)$

Score of the 10 overs in ODI

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

- Runs scored in $7^{th}$ over

= no of runs in overs [7, 7]

= score(7) − score[6] = 65 − 49 = 16

2 8 6 15

1st $2^{nd}$ $3^{rd}$ $4^{th}$

- Runs scored in $6^{th}$ to $10^{th}$ over

runs in [6, 10]

score[10] − score[5]

97 − 31 = 66

- Runs scored in $10^{th}$ over

[10, 10]

= score[10] − score[9] = 97 − 88 = 9

- Runs in $3^{rd}$ to $6^{th}$ over

score[6] − score[2]

= 49 − 8 = 41

- Runs in $4^{th}$ to $9^{th}$ over

= score[9] − score(3)

88 − 14 = 74

A: -3 6 2 4 5 2 8 -9 3 1
0 1 2 3 4 5 6 7 8 9

Pf -3 3 5 9 14 16 24

Prefix Sum is sum till that point
(index)

Pf[i] → Sum of all elem from
index 0 till index i

PF[0] = ar[0]
PF[1] = ar[0]+ar[1] = pf[0]+a[1]

PF[2] =

PF[3] =

PF[i] = pf[i-1] + ar[i]
PF[0] =

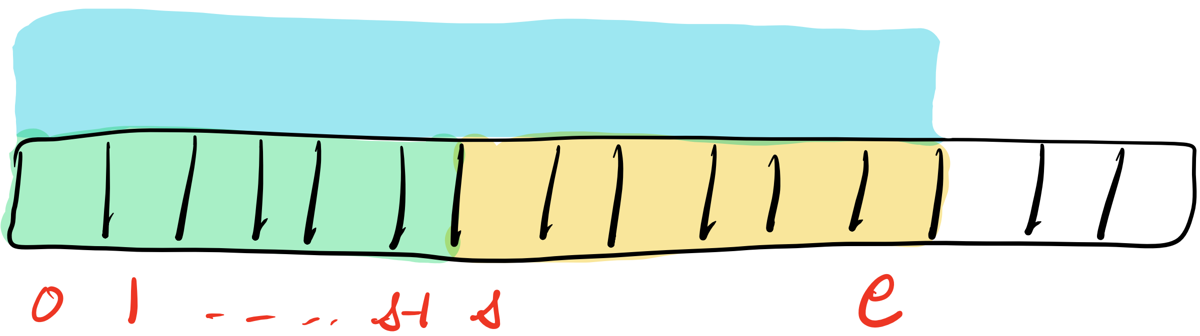
// Build PS            int pf[n]

First step ??

```
pf[0] = ar[0]
for(i=1; i<n; i++){
    pf[i] = pf[i-1] + a[i]
}
```

TC: $O(n)$            SC: $O(n)$

- What is the sum of all no.s in range [s,e] using pf array.





pf[e] = pf[s-1] + sum(s,e)

sum(s,e) = pf[e] − pf[s-1]

To answer the Q queries

1) Create Pf array $TC = O(N)$

2) For each query [s,e], $TC = O(Q)$

$$sum = pf(e) - pf(s-1)$$

Total $TC = O(N+Q)$

$SC = O(N)$

| | |
|---|---|
| TC: $O(NQ)$ | $TC = O(N+Q)$ |
| SC: $O(1)$ | $SC = O(N)$ |

$N = 10^5$ $Q = 10^5$

$N \times Q = 10^{10}$ ~ 100 sec

$N + Q = 2 \times 10^5$ 2 milliseconds

$$\frac{2 \times 10^5}{10^8} = \frac{2}{10^3}$$

```
for (i=0 ; i< Q ; i++) {
        // s, e
    print ( pf[e] - pf[s-1] )
}
```

$O(Q)$

Q3 Given an array and Q queries

DirectI

Flipkart

s, e, O ⇒ sum of all odd index elem in range [s, e]

s, e, E ⇒ sum of all even index elem in range [s, e]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A: | 2 | 3 | 1 | -1 | 0 | 8 | 5 | 4 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Q=2

| s | e | O/E | |
|---|---|---|---|
| 3 | 6 | O | 7 |
| 1 | 5 | E | 1 |

Idea: Can we create something like $Pf_{odd}$ and $Pf_{even}$ ?

$Pf_{odd}[i]$ = sum of all odd indexed elements till i

$Pf_{even}[i]$ = sum of all even indexed elements till i

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | 2 | 3 | 1 | 6 | 4 | 5 |
| $Pf_{even}$ | 2 | 2 | 3 | 3 | 7 | 7 |

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| A: | 2 | 4 | 3 | 1 | 5 |
| Pfeven | 2 | 2 | 5 | 5 | 10 |

Sum of even idx elem in [s,e] =

$Pf_{even}[e] - Pf_{even}[s-1]$

Sum of odd elem in [s,e] =

$Pf_{odd}[e] - Pf_{odd}[s-1]$

**Code**

break back at 10:20

1) $Pf_{even}[n]$

$Pf_{even}[0] = ar[0]$

```
for ( i=1 ; i<n ; i++) {
        if ( i%2 == 1) {            //odd
            pfe[i] = pfe[i-1]
        }
        else {
            pfe[i] = pfe[i-1] + ar[i]
}
```

Q3 Count the number of Special Index

Codenation
JP Morgan
Directi

Special idx is : after removing
sum of even idx elem = sum of odd idx

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| A: | 4 | 3 | 2 | 7 | 6 | -2 |

| i | 0 | 1 | 2 | 3 | 4 | Se | So | |
|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 2 | 7 | 6 | -2 | 8 | 8 | ✓ |
| 1 | 4 | 2 | 7 | 6 | -2 | 9 | 8 | × |
| 2 | 4 | 3 | 7 | 6 | -2 | | | |
| 3 | 4 | 3 | 2 | 6 | -2 | | | |

Quiz

| 0 | 1 | ~~2~~ | 3 | 4 |
|---|---|---|---|---|
| 4 | 1 | 3 | 7 | 10 |

| 0 | 1 | 2 | 3 |
|---|---|---|---|
| 4 | 1 | 7 | 10 |

= 11

**Quiz**

X

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 1 | 4 | 0 | -1 | 2 | -2 | 10 | 8 |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 1 | 0 | -1 | 2 | -2 | 10 | 8 | = 15 |

**Quiz**

2 3 1 4 0 -1 2 -2 10 8

**Obs:** After removing $i$

$$\underbrace{0, 1, 2, 3, \ldots, i-1}, \; i, \; \underbrace{i+1, \ldots, n-1}$$

| before | after | before | after |
|---|---|---|---|
| even | even | even → | odd |
| odd | odd | odd → | even |

After removing $i$.

Sum of even idx =

sum of even $[0, i-1]$ +

sum of odd $[i+1, n-1]$

Sum of odd idx =

sum of odd $[0, i-1]$ +

sum of even $[i+1, n-1]$

Pseudo code

```
count = 0
for (i 0 → n-1) {
    Se = sum of even after remove i
    So = sum of odd after remove i
    if ( Se == So )
        count ++
}
return count
```

TC : O(n)    SC : O(n)

{don}

1) Carry fwd technique
2) Subarrays