SCALER

Array - prefix sum

TABLE OF CONTENTS

Practical Scenrio

You want to develop a feature for a banking app that allow users to quickly access their stock portfolio's profit or loss over specific periods.

arr → profit or loss from a particular stock over a period of days.

arr = [-5, 10, 20, 40, 50, -10, 80, -90, -20, -10]

-5	10	20	40	50	-10	80	-90	-20	-10
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
0	1	2	3	4	5	6	7	8	9

start	end.	ans.
0	9	65
1	4	120
0	0	-5
2	7	90

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

Note : L and R are indices such that L ≤ R. $1 \le N, Q \le 10^5$

arr[10] → [-3 6 2 4 5 2 8 -9 3 1]
0 1 2 3 4 5 6 7 8 9

Queries

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

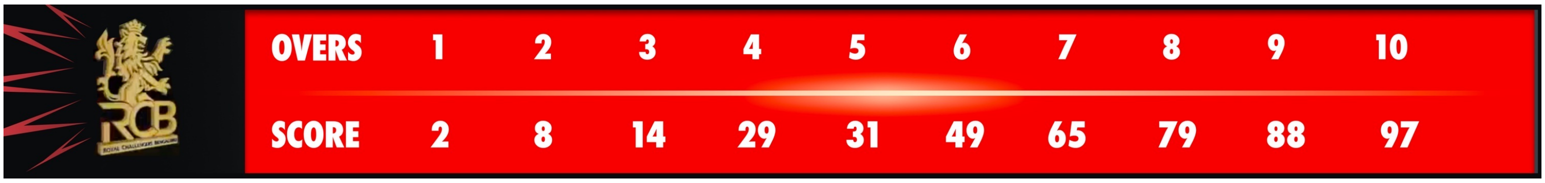
Brute force Approach →

For every query,
iterate from L to R and find the sum.

```
for ( i = 0; i < Q ; i++) {
        L → Q [i][0]
        R → Q [i][1]

        sum = 0;

        for( j = L ; j ≤ R ; j++){
                sum += arr[j];
        }

        print(sum);
}
```

T.C → O(Q*N)
S.C → O(1)
↓
T.L.E

- **Given Royal Challengers Bengaluru's cricket scores for first 10 overs of batting.**



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

⇓

Cumilative Sum / prefix Sum

- Runs scored in 7th over = 65 - 49 = 16
- Runs scored from 6 - 10th over = 97 - 31 = 66
- Runs scored in 10th over = 97 - 88 = 9
- Runs scored from 3 - 6th over = 49 - 8 = 41
- Runs scored from 4 - 9th over = 88 - 14 = 74
- Runs scored in lth - rth over = runs[r] - runs[l-1]

How to create psum()

arr[10] → [-3 6 2 4 5 2 8 -9 3 1]

0 1 2 3 4 5 6 7 8 9

psum[10] = [-3, 3, 5, 9, 14, 16, 24, 15, 18, 19]

0 1 2 3 4 5 6 7 8 9

Query.

4	8
3	7
1	3
0	4
7	7

0 → psum[8] - psum[4-1] ⇒ 9

1 → psum[7] - psum[3-1] ⇒ 10

2 → psum[3] - psum(1-1) ⇒ 12

3 → psum[4] ⇒ 14

4 → psum[7] - psum[7-1] ⇒ -9

arr[6] → [10 32 6 12 20 1]

0 1 2 3 4 5

psum[10] = (10, 42, 48, 60, 80, 81]

code. to create psum()

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

</> Code

```
//1. Create pSum[]

    int pSum[N];
      pSum[0] = arr[0];
      for( i=1; i<N; i++){
            pSum[i] = pSum[i-1] + arr[i];
      }

//2. Answer the queries
```

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

```
      for( i=0; i<Q; i++){
            L -> Q[i][0];
            R -> Q[i][1];

            if( L==0) {
                  print( pSum[R];
            }
            else {
                  print( pSum[R] - pSum[L-1]);
            }
      }
```

T.C → $O(N+Q)$

S.C → $O(N)$

Modification of same array into psum

arr[10] → [-3 6 2 4 5 2 8 -9 3 1]

0 1 2 3 4 5 6 7 8 9

↓

arr[10] → [-3, 3, 5, 9, 14, 16, 24, 15, 18, 19]

```
for( i = 1; i < N; i++){
        arr[i] = arr[i] + arr[i-1];
}
```

S.C → O(1)

< Question > : Given an arr[N] and Q queries with start(s) and end(e) index. For every query print sum of all even indexed elements from s to e.

arr[ ] →	2	3	1	6	4	5
	0	1	2	3	4	5

Queries

s	e		
1	3	→	1
2	5	→	5
0	4	→	7
3	3	→	0

[BF Idea] - For every query, iterate from s to e.

[T.C → O(N×Q)
S.C → O(1)]

[Idea] - *Using prefix sum* of even. indexed elements only.

arr[ ] →

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

psum[10] →

0	1	2	3	4	5
2	2	3	3	7	7

Queries

s	e	
1	3	→ 1
2	5	→ 5
0	4	→ 7
3	3	→ 0

psum[e] - psum[s-1]

< / > Code

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

N → 100

Q → 10^5

// → todo. (similar to previous ques)

T.C → O(N+Q)

S.C → O(N)

modify array → O(1)

< Question > : For all the queries, find the sum of odd indexed elements from s to e.

0 1 2 3 4 5

⇒ todo.

Special Index

< Question > : Given an arr[N], count the number of special indices in the array.

Special Index : Index after removing which,

Sum of even indexed elements = sum of odd indexed elements.

$1 \leq N \leq 10^5$

arr = $[\underset{0}{4},\ \underset{1}{3},\ \underset{2}{2},\ \underset{3}{7},\ \underset{4}{6},\ \underset{5}{-2}]$

ans = 2.

remove idx = 0 → $[\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{7}\ \ \underset{4}{6}\ \ \underset{5}{-2}] \rightarrow [\underset{0}{3}\ \ \underset{1}{2}\ \ \underset{2}{7}\ \ \underset{3}{6}\ \ \underset{4}{-2}]$ ✓

remove idx = 1 → $[\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{7}\ \ \underset{4}{6}\ \ \underset{5}{-2}] \rightarrow [\underset{0}{4}\ \ \underset{1}{2}\ \ \underset{2}{7}\ \ \underset{3}{6}\ \ \underset{4}{-2}]$ ✗

remove idx = 2 → $[\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{7}\ \ \underset{4}{6}\ \ \underset{5}{-2}] \rightarrow [\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{7}\ \ \underset{3}{6}\ \ \underset{4}{-2}]$ ✓

remove idx = 3 → $[\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{7}\ \ \underset{4}{6}\ \ \underset{5}{-2}] \rightarrow [\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{6}\ \ \underset{4}{-2}]$ ✗

remove idx = 4 → $[\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{7}\ \ \underset{4}{6}\ \ \underset{5}{-2}] \rightarrow [\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{7}\ \ \underset{4}{-2}]$ ✗

remove idx = 5 → $[\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{7}\ \ \underset{4}{6}\ \ \underset{5}{-2}] \rightarrow [\underset{0}{4}\ \ \underset{1}{3}\ \ \underset{2}{2}\ \ \underset{3}{7}\ \ \underset{4}{6}]$ ✗

Quiz

1.

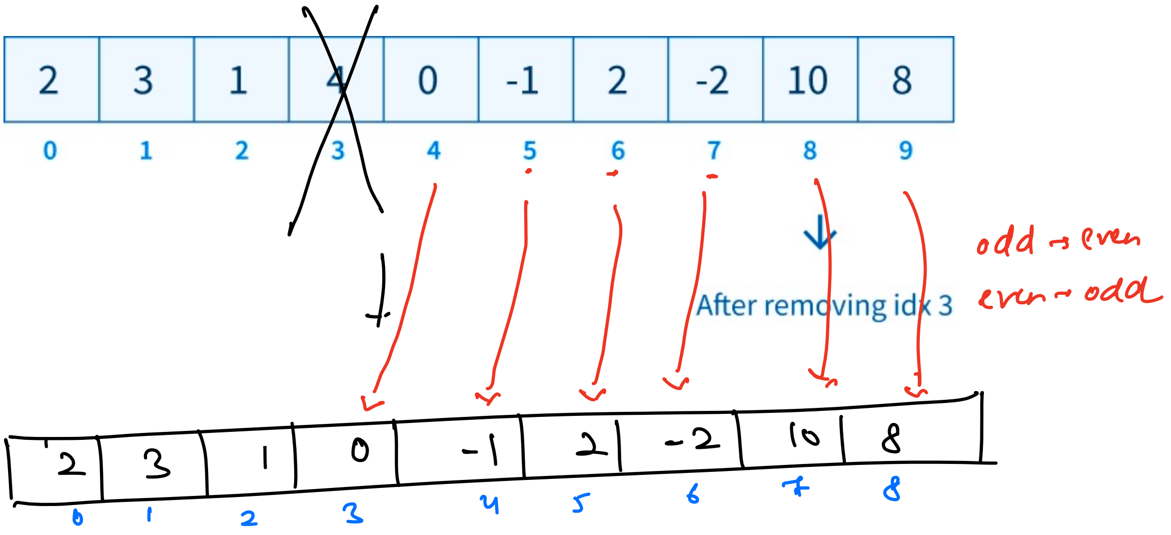
4	1	3	7	10
0	1	2	3	4

→

4	1	7	10
0	1	2	3

Sum of odd indexed elements → 11

2.



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

After removing idx 3

odd → even
even → odd

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

$$S_{o[0-9]} = S_{o[0-2]} + S_{e[4-9]}$$

$$(3) + (12) = \underline{15}$$

$S_{e[0-9]}$ after removing 3rd index → $S_{e[0-2]} + S_{o[4-9]}$

→ $\underline{8}$.

[Idea - 2]

arr = [2, 3, 1, 4, 0, -1, 2, -2, 10, 8]

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

Delete 4th index

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

$S_{o[0-9]}$ after deleting 4th index → $S_{o[0-3]} + S_{e[5-9]}$

→ $\underline{19}$.

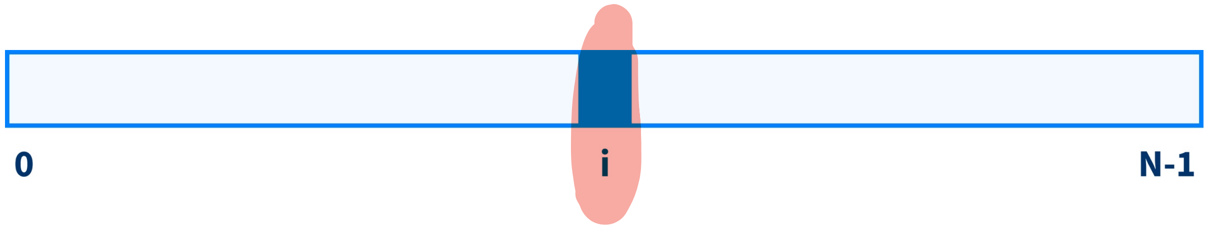
$S_{e[0-9]}$ after deleting 4th index → $S_{e[0-3]} + S_{o[5-9]}$

→ $\underline{8}$

pfEven[ ]

pfOdd[ ]

Assumption, we already have pfeven[ ] and pfodd[ ].



sum of even indexed elements after removing ith indexed element →

$$\underline{S_e[0, i-1]} + \underline{S_o[i+1, N-1]}$$

$$\downarrow \qquad\qquad\qquad \downarrow$$

$$pfEven[i-1] + pfodd[N-1] - pfodd[i]$$

sum of odd indexed elements after removing ith indexed element →

$$\underline{S_o[0, i-1]} + \underline{S_e[i+1, N-1]}$$

$$\downarrow \qquad\qquad\qquad \downarrow$$

$$pfodd[i-1] + pfEven[N-1] - pfEven[i]$$

</> Code

```
// Create pfEven[] and pfodd[]

pfEven[N],  pfodd[N];

pfEven[0] = arr[0];

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

pfodd[0] = 0;

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

```
int count = 0;

if( pfodd[N-1] - pfodd[0] == pfEven[N-1] - pfEven[0]){
        count++;
}

for( i=1; i<N; i++){
        Se = pfEven[i-1] + pfodd[N-1] - pfodd[i];
        So = pfodd[i-1] + pfEven[N-1] - pfEven[i];

        if( Se == So) { count++ }
}

return count;
```

T.C → O(N)
S.C → O(N)

→ Carry Forward

→ Basic of Subarrays.

→ optimisation → Sum of elements is req. again & again.

arr [7, 2, 8, 3, 8, 6, 8]

↓

$1 + 6 + 0 + 5 + 0 + 2 + 0 = \underline{\underline{14}}$

① find max element

② Keep on taking diff. with max element.