Array - prefix sum

TABLE OF CONTENTS

< 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 ≤ N, Q ≤ 10^5

arr[10] → [-3 6 2 4 5 2 8 -9 3 1]

0 1 2 3 4 5 6 7 8 9

Pseudocode:

```
Simple solution =>
    for each query :
        l, r
        sum = 0
        for( i : l → r)
            sum + = arr[i]
        print (sum)
```

Queries

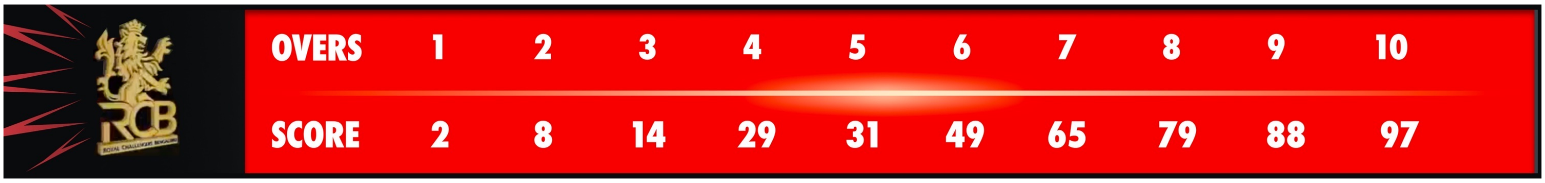
L	R	
4	8	=> 9
3	7	=> 10
1	3	=> 12
0	4	=> 14
7	7	=> -9

1 query => N

Q queries => NQ

TC : O(NQ)

- **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

- Runs scored in 7th over = runs[7] - runs[6] = 65-49
  = 16
- Runs scored from 6 - 10th over = runs[10] - runs[5] = 97-31
  = 66
- Runs scored in 10th over = runs[10] - runs[9] = 97-88
  = 9
- Runs scored from 3 - 6th over = runs[6] - runs[2] = 49-8
  = 41
- Runs scored from 4 - 9th over = runs[9] - runs[3] = 88-14
  = 74
- Runs scored in lth - rth over = runs[r] - runs[l-1]

  l-r

Prefix sum

How to create psum()

10 32 6 12 20 1

⇒ 10 42 48 60 80 81

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

10 42 48 60 80 81

arr[6] → [10 32 6 12 20 1]

psum[6] = [10 42 48 60 80 81

arr[N] len(arr)

1) Simple idea

create pf[N]

pf[i] ⇒ a[0] + a[1] + a[2] . . . . a[i]

</> Code

```
for i 0 → n-1
    sum = 0
    for j 0 → i
        sum += arr[j]
    pf[i] = sum
```

$pf[0] = arr[0]$

$pf[1] = arr[0] + arr[1] = pf[0] + arr[1]$

$pf[2] = arr[0] + arr[1] + arr[2] = pf[1] + arr[2]$

$pf[3] = arr(0) + arr(1) + arr(2) + arr(3) = pf[2] + arr[3]$

$\vdots$

$$pf[i] = \underbrace{arr[0] + arr[1] + \dots + arr[i-1]}_{pf[i-1]} + arr[i] = pf[i-1] + arr[i]$$

```
pf[N]
pf[0] = arr[0]
for (i : 1 → n-1)
    pf[i] = pf[i-1] + arr[i]
```

TC: $O(N)$

Optimisation for 1st question

By creating prefix arrays

Step 1 Create Prefix sum array

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

TC: O(N)

SC: O(N)

Queries - 5

L	R
4	8
3	7
1	3
0	4
7	7

→ pf[8] - pf[3] = 18 - 9 = 9

→ pf[7] - pf[2] = 15 - 5 = 10

→ pf[3] - pf[0] = 9 - (-3) = 12

→ pf[4] = 14

→

For sum from l to r

</> Code

```
if (l==0)
        pf[r]
else
        pf[r] - pf[l-1]
```

Modification of same array into psum

0 1 2 3 4 5 6 7 8 9

10	~~20~~ 30	~~30~~ 60	~~40~~ 100
0	1	2	3

```
arr[1] += arr[0]
arr[2] += arr[1]
arr[3] += arr[2]
```

```
for i: 1 → n-1
    arr[i] += arr[i-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] -

Array - prefix sum

[Idea] - ***Using prefix sum***

2	4	3	1	5
0	1	2	3	4
2	2	5	5	10

arr[ ] →

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

psum[~~10~~6] →
even

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

</> Code

```
pf[N]

pf[0] = arr[0]
for( i: 1→n-1)
        if (i%2 == 0)
                pf[i] = pf[i-1] + arr[i]
    else
            pf[i] = pf[i-1]
```

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

0 1 2 3 4 5

```
pf[0] = 0
for( i:  1 → n-1)
        if (i%2 != 0)
                pf[i] = pf[i-1] + arr[i]
        else
                pf[i] = pf[i-1]
```

Codenation, GS, Microsoft, Cisco

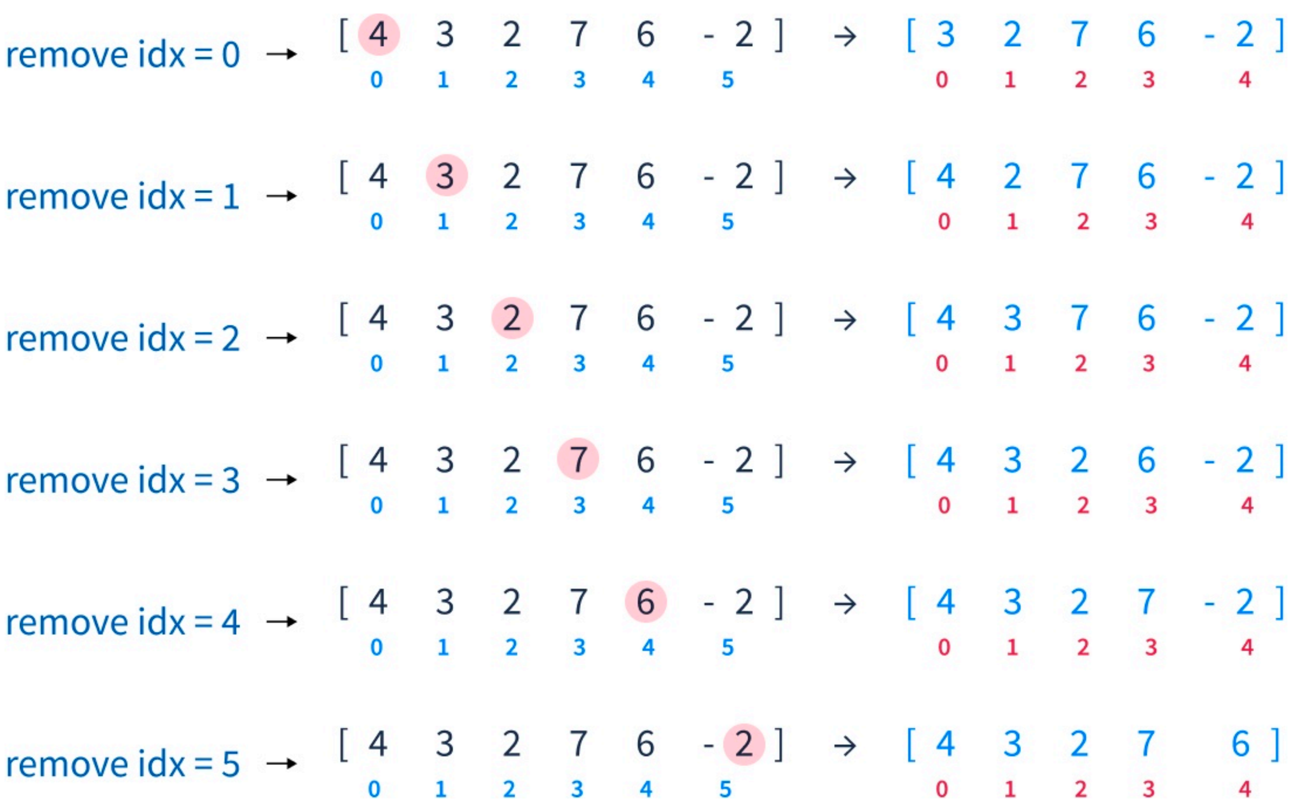
Special Index

< 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.

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



remove idx = 0 → [4 3 2 7 6 -2] → [3 2 7 6 -2]
0 1 2 3 4 5 — 0 1 2 3 4

remove idx = 1 → [4 3 2 7 6 -2] → [4 2 7 6 -2]
0 1 2 3 4 5 — 0 1 2 3 4

remove idx = 2 → [4 3 2 7 6 -2] → [4 3 7 6 -2]
0 1 2 3 4 5 — 0 1 2 3 4

remove idx = 3 → [4 3 2 7 6 -2] → [4 3 2 6 -2]
0 1 2 3 4 5 — 0 1 2 3 4

remove idx = 4 → [4 3 2 7 6 -2] → [4 3 2 7 -2]
0 1 2 3 4 5 — 0 1 2 3 4

remove idx = 5 → [4 3 2 7 6 -2] → [4 3 2 7 6]
0 1 2 3 4 5 — 0 1 2 3 4

Quiz

1.

4	1	3	7	10
0	1	2	3	4

→

4	1	7	10
0	1	2	3

Sum of odd after removal of idx 2

= sum of odd idx before 2

+ sum of even idx after 2

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

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

[BF Idea] -

</> Code

After removal of idx i

odd_sum ⇒ odd sum before i +
even sum after i

even_sum ⇒ even sum befor i +
odd sum after i

```
if ( odd_sum = even_sum )
        count ++
```

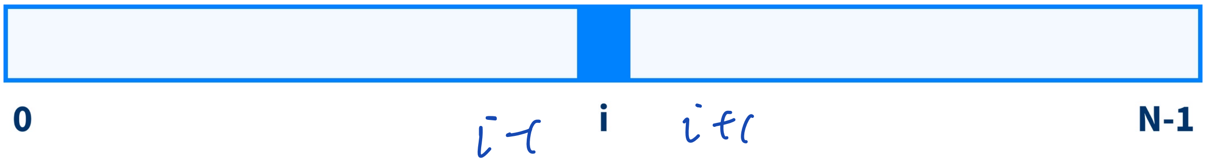
[Idea - 2]

0 1 2 3 4 5 6 7 8 9

Delete 4th index

0 1 2 3 4 5 6 7 8

Assumption, we already have peven[ ] and podd[ ].



sum of even indexed elements after removing ith indexed element →

pf_even [i-1] + pf_odd [n-1] − pf_odd [i]

sum of odd indexed elements after removing ith indexed element →

pf_odd [i-1] + pf_even (i+1, n-1)

↓

pf_even (n-1) − pf_even (i)

odd_sum ⇒ odd sum before i +
even sum after i

even_sum ⇒ even sum befor i +
odd sum after i

< / > Code