Arrays - Prefix Sum

Question

Given N elements & Q queries. For each query, calculate sum of elements from L to R (0-based index) $L <= R$

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

Q = 6

L	R	
4	8	= 5+2+8+-9+3 = 9
3	7	= 10
1	3	= 12
0	4	
6	9	
7	7	= -9

Brute force soln

```
for(i=0; i<Q; ++i) {        → Q times
    // for each query, l & r are given
    sum=0;
    for(j=l; j<=r; ++j) {   → [1, N] times
        sum += a[j]
    }
    print(sum)
}
```

TC: O(N × Q)

SC: O(1)

constraints: $N <= 10^5$, $Q <= 10^5$

Above code will give TLE

Given Indian team scores for first 10 overs of batting.

After every over, current score is given.

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

1. Runs scored in just 7^{th} over

= 65 - 49 = 16

2. Total runs scored in last 5 overs 6th to 10th over.

score [6-10]

score [1-10] = score [1-5] + score [6-10]

score [6-10] = score [1-10] - score [1-5]

= 97 - 31 = 66

3. Total runs scored in last over

score [10-10] = 97 - 88 = 9

4. 3rd - 6th over

= 49 - 8 = 41

5th 4th - 9th over

= 88 - 14 = 74

If we have a cumulative array, we can answer the range queries faster.

pf[n] cumulative sum

pf[i] → sum of all elements from 0 to i index.

	0	1	2	3	4
a[] =	2	5	-1	7	1
pf[] =	2	7	6	13	14

a[] =	10	32	6	12	20	1
pf[] =	10	42	48	60	80	81

pf[n] → SC: O(N)

```
for (i=0; i<n; ++i) {
    sum=0;
    for(j=0; j<=i; ++j){
        sum += a[j]
    }
    pf[i] = sum
}
```

pf[i] = sum of a[0] to a[i]

TC: $O(N^2)$

SC: O(N)

pf[0] = a[0]

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

pf[2] = a[0] + a[1] + a[2] = pf[1] + a[2]

pf[3] = a[0] + a[1] + a[2] + a[3] = pf[2] + a[3]

pf[i] = pf[i-1] + a[i]

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

TC : O(N)
SC : O(N)

Coming back to Question 1

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

sum[L,R] = ?

sum[0,R] = pf[R]

sum[0,R] = sum[0, L-1] + sum[L,R]

sum[L,R] = sum[0,R] - sum[0, L-1]

= pf[R] - pf[L-1]

Code

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

TC : O(N+Q)

SC : O(N)

```
for (i=0; i<Q; ++i) {  → Q times
    // sum of elements from l to r
    if (l==0) {
        sum = pf[r]
    }
    else {
        sum = pf[r] - pf[l-1]
    }
    print(sum)
}
```

Question

Given a array N & Q queries. For every query find sum of all even-indexed elements from l to r.

	0	1	2	3	4	5
A[] =	2	3	1	6	4	5

Q=4

1 3 = 1

2 5 = 1+4=5

0 4 = 2+1+4=7

3 3 = 0

	0	1	2	3	4	5
A[] =	2	3	1	6	4	5
pf_e[] =	2	2	3	3	7	7

pf[i] = pf[i-1] + a[i] if i is even

pf[i] = pf[i-1] + 0 if i is odd

```
pfe[0] = a[0]
for (i=1; i<n; ++i) {
    if (i%2 == 0)
        pfe[i] = pfe[i-1] + a[i]
    else
        pfe[i] = pfe[i-1]
}
```

A[] = 2 4 3 1 5

Pf[] = 2 2 5 5 10

Prefix sum of odd indexed elements

$pf[i] = pf[i-1] + a[i]$ if i is odd

$pf[i] = pf[i-1] + 0$ if i is even

```
pf[0] = a[0] 0
for (i=1; i<n; ++i) {
    if (i%2 != 0)
        pf[i] = pf[i-1] + a[i]
    else
        pf[i] = pf[i-1]
}
```

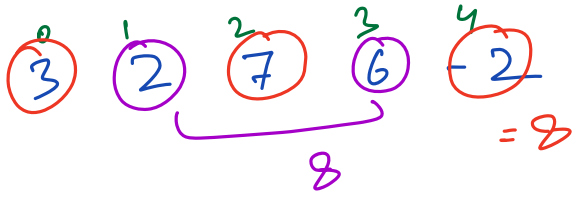
Question

Given an array of size N, count no. of special index in the array.

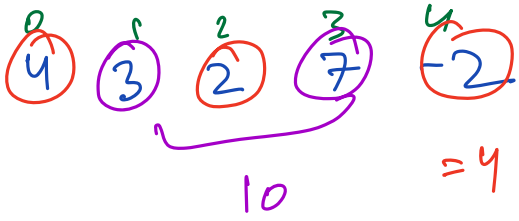
Note: special index is after removing it, sum of even indexed elements = sum of odd indexed elements

a[6] = 4 3 2 7 6 -2 (indices 0 1 2 3 4 5)

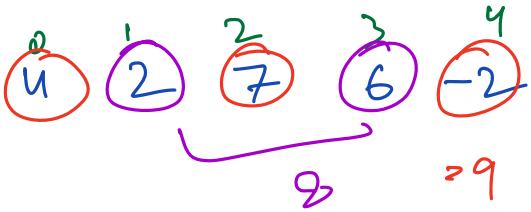
delete 0th index



delete 4th index



delete 1st index



Similar for rest of indices

A = [4 1 3 7 10] (indices 0 1 2 3 4)

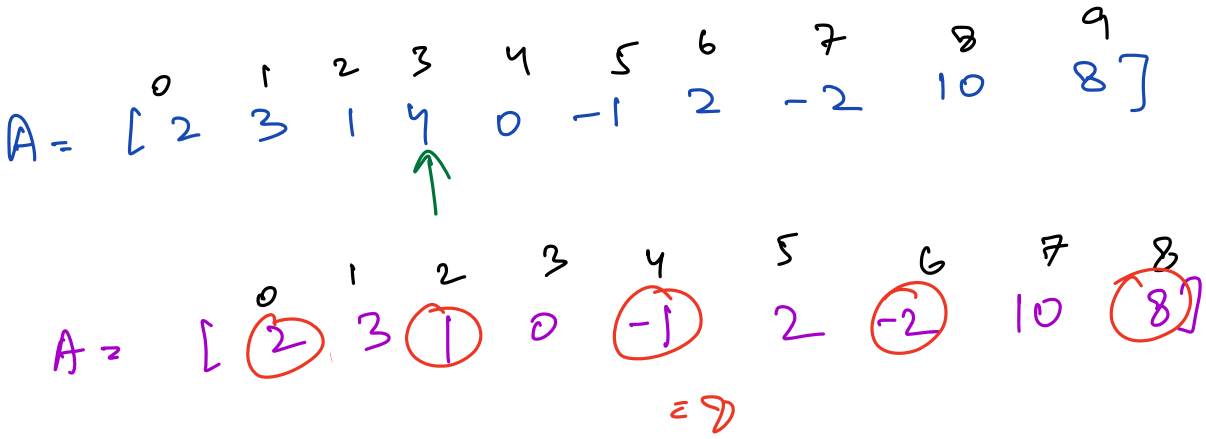
A = 4 1 7 10 (indices 0 1 2 3)

= 11

A = [2 3 1 4 0 -1 2 -2 10 8] (indices 0 1 2 3 4 5 6 7 8 9)

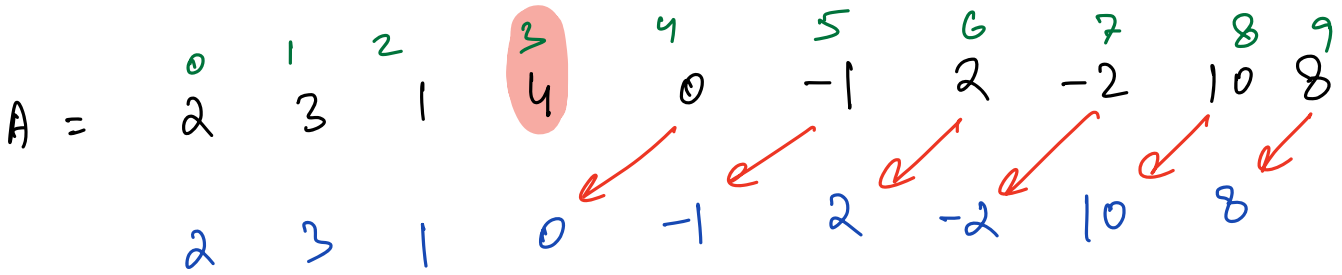
A = [2 3 1 0 -1 2 -2 10 8] (indices 0 1 2 3 4 5 6 7 8)

= 15



Suppose we want to check if i is special index?

→ Indices on left side of i will remain intact while indices on right side will change from odd ↔ even



Sum of odd-indexed elements after removing index 3 =

Sum of odd-indexed elements from 0 to 2 (=3) +

Sum of even-indexed elements from 4 to 9

= 0 + 2 + 10 = 12

sum of even-indexed elements after removing index 3 =

sum of even-indexed elements from 0 to 2 + (= 2+1)

sum of odd-indexed elements from 4 to 9

= -1 + -2 + 8

Create pfe & pfo array → for odd-indexed

↳ prefix sum of even-indexed elements

for checking index i

S_{odd} after removing = $S_{odd}[0, i-1]$ + $S_{even}[i+1, n-1]$

= $pfo[i-1]$ + $(pfe[n-1] - pfe[i])$

S_{even} after removing = $S_{even}[0, i-1]$ + $S_{odd}[i+1, n-1]$

= $pfe[i-1]$ + $(pfo[n-1] - pfo[i])$

```
int specialCount (a[]){
    pfe[n], pfo[n]  → create TODO        O(N): TC
                                         O(N): SC
    c=0
    for(i=0; i<n; ++i) {
        if(i==0) {
            so = pfe[n-1] - pfe[i]
            se = pfo[n-1] - pfo[i]
        } else {
            so = pfo[i-1] + pfe[n-1] - pfe[i]
            se = pfe[i-1] + pfo[n-1] - pfo[i]
        }
        if(so == se)
            ++c
    }
    return c;
}
```

TC: O(N)

SC: O(N)