Prefix Sum

Question :- Given N elements & Q queries. For each query, calculate sum of all elements from L to R [0 based index].

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

Queries :-

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

Brute force

:- For each query go from L → R

PSEUDO CODE

```
void querySum( Queries[][] , A [], querySize , size )
    for ( i = 0 ; i < querySize; i++){
        L = Queries[i][0];
        R = Queries[i][1];

        sum = 0
        for ( j = L ; j <= R ; j++){
            sum += A[j];
        }
```

}

}

Print (Sum);

$TC = O(Q * N)$

$SC = O(1)$

$\boxed{\text{Constraints :-} \quad 1 <= N <= 10^5 \quad 1 <= Q <= 10^5}$

$\therefore$ Total iteration $\approx 10^{10} \Leftarrow \boxed{TLE}$

Quiz 1 :- Given the Scores of the 10 overs of a cricket match.

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

How many runs were scored in just 7th over?

Ans :- 65 − 49 = 16

Quiz 2 :- Given the Scores of the 10 overs of a cricket match.

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

How many runs were scored from 6th to 10th over [both included]

Ans = 97 − 31 = 66

Quiz 3 :- Given the Scores of the 10 overs of a cricket match.

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

How many runs were scored in just 10^{th} over?

Ans $\Rightarrow$ $97 - 88 = 9$

Quiz 4 :- Given the Scores of the 10 overs of a cricket match.

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

How many runs were scored from 3^{rd} to 6^{th} over [both included]

Ans $= 49 - 8 = 41$

Quiz 5 :- Given the Scores of the 10 overs of a cricket match.

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

How many runs were scored from 4^{th} to 9^{th} over [both included]

Ans $= 88 - 14 = 74$

OBSERVATION for OPTIMIZATION

① On Observing the cricket board Score, we can say that queries in just constant time since we have CUMMULATIVE SCORES.

② In Similar manner, if we have Cummulative Sum array for the above problem, we should be able to answer in just constant Time.

③ Hence we need CUMMULATIVE SUM or PREFIX ARRAY for above problem.

PREFIX SUM ARRAY

PS[i] = Sum of all the element from 0 to i^{th} index

Ex:- arr = [2, 5, -1, 7, 1]

PS = [2, 7, 6, 13, 14]

Quiz 6 :- Calculate the prefix sum array for following array :-

arr[] = 10, 32, 6, 12, 20, 1

PS = 10 42 48 60 80 81

PSEUDO CODE

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

Observation for optimizing prefix sum

(1)

pf[0] = A[0];
pf[1] = A[0] + A[1];
pf[2] = A[0] + A[1] + A[2]
pf[3] = A[0] + A[1] + A[2] + A[3];
pf[4] = A[0] + A[1] + A[2] + A[3] + A[4];

↳ Lot of Redundant Code

So, I can write.

pf[0] = A[0]
pf[1] = pf[0] + A[1]
pf[2] = pf[1] + A[2]
pf[3] = pf[2] + A[3]
pf[4] = pf[3] + A[4]

⇒ Generically ⇒ pf[i] = pf[i-1] + A[i]

PSEUDO CODE

```
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) ⇒ we are creating a Prefix Sum array

Initial problem

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

Queries:-

L	R	Solution
4	8	pf[8] - pf[3] = 18 - 9 = 9
3	7	pf[7] - pf[2] = 15 - 5 = 10
1	3	pf[3] - pf[0] = 9 - (-3) = 12
★★★ 0	4	pf[4] = 14
7	7	pf[7] - pf[6] = 15 - 24 = -9

Generalisation

Sum [L to R] = Pf [R] − Pf[L−1] (L != 0)
Pf [R] (L == 0)

PSEUDO CODE

```
void querySum( Queries[][] , A [],
               querySize , size )
    Pf [N];
    pf[0] = A[0];
    for ( i = 1 ; i < N ; i++){
        Pf[i] = Pf[i-1] + A[i];
    }

    for ( i = 0 ; i < querySize; i++){
        L = Queries [i][0];
        R = Queries [i][1];
        Sum = 0
        if (L == 0) {
            Sum = Pf [R];
        } else {
            Sum = Pf [R] − Pf[L-1];
        }

        Print (Sum);
    }
}
```

}

$TC = O(N + Q)$ — creating Pf (N), finding ans for queries (Q)

Constraints :

$1 <= N <= 10^5$
$1 <= Q <= 10^5$

$TC \rightarrow 10^5 + 10^5 \approx 2 * 10^5$

$SC \Rightarrow O(N) \rightarrow$ creating PF array

Question :- Can you optimize space?

Ans ⇒ By using the given array as pf array.

Ex:- $A[] = [-3, \overset{3}{\cancel{6}}, \overset{5}{\cancel{2}}, \overset{9}{\cancel{4}}, \overset{14}{\cancel{5}}, \overset{16}{\cancel{2}}, \overset{24}{\cancel{8}}, \overset{15}{\cancel{-9}}, \overset{18}{\cancel{3}}, \overset{19}{\cancel{1}}]$

PSEUDO CODE

```
void PfInplace ( A[]) {
    for (i = 1; i < A.length; i++){
        A[i] = A[i-1] + A[i];
    }
}
```

Question :- Sum of even indexed element from L to R.

Ex :- arr[] = { 2 3 1 6 4 5 } (indices 0 1 2 3 4 5)

Queries

L	R		Sum
1	3	→	1
2	5	→	5
0	4	→	7
3	3	→	0

Brute force

For each query, go from L to R to get the sum

TC ⇒ $O(Q * N)$

OPTIMIZED

① what if we create prefix sum array of even indexes

Ex ⇒ arr[] = { 2, 3, 1, 6, 4, 5 } (indices 0 1 2 3 4 5)

Pf_e = { 2, 2, 3, 3, 7, 7 }

PSEUDO CODE:

```
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]
}
}

Quiz 7 :- Construct the prefix sum for even indexed elements for the given array.

arr[] = [2, 4, 3, 1, 5]

(indices 0 1 2 3 4)

PS[] = [2, 2, 5, 5, 10]

⇒ Move back to Original question

Ex :- arr[] = { 2 3 1 6 4 5 }

(indices 0 1 2 3 4 5)

Pfe[] = { 2 2 3 3 7 7 }

Queries

L	R	Sum
1	3	Pfe[3] - Pfe[0] = 3-2 = 1
2	5	Pfe[5] - Pfe[1] = 7-2 = 5
0	4	7
3	3	0

PSEUDO CODE

```
void querySum( Queries[][] , A[],
               querySize , size )
    Pfe[N];
    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];
```

```
        }
    }
    for (i=0 ; i < querySize; i++){
        L = Queries[i][0];
        R = Queries[i][1];
        Sum = 0
        if (L == 0) {
            Sum = Pf_e[R];
        } else {
            Sum = Pf_e[R] - Pf_e[L-1];
        }
        Print(Sum);
    }
}
```

TC = O(N+Q)

SC = O(N)

EXTENSION :- For Odd Indexes.

	0	1	2	3	4	5
arr[] =	{2	3	1	6	4	5}
Pf_o =	{0	3	3	9	9	14}

PSEUDO CODE

```
Pf_o[N];
Pf_o[0] = 0;
for (i -> 1 to N-1){
        if i is odd;
            Pf_o[i] = Pf_o[i-1] + A[i]
        else:
            Pf_o[i] = Pf_o[i-1]
}
```

Question 2 :- Given an array of N, count the number of special index in array.

Note :- Special Indices are those after removing which, sum of all EVEN index elements is equal to sum of all ODD indexed elements

Ex:- arr = { 4, 3, 2, 7, 6, -2 }

(indices 0, 1, 2, 3, 4, 5 written above the elements)

i	Array	Se	So	Special
0	{3, 2, 7, 6, -2}	8	8	yes
1	{4, 2, 7, 6, -2}	9	8	No
2	{4, 3, 7, 6, -2}	9	9	yes
3	{4, 3, 2, 6, -2}	4	9	No
4	{4, 3, 2, 7, -2}	4	10	No
5	{4, 3, 2, 7, 6}	12	10	No

$\boxed{\text{Ans} = 2}$

Quiz 8 :- what will be the sum of elements at ODD indices in the Resulting array after removal of index 2 ?

A[] = [4, 1, 3, 7, 10]

(indices 0, 1, 2, 3, 4 written above the elements)

Ans A[] = [4, 1, 7, 10] ⇒ 11

(indices 0, 1, 2, 3 written above the elements)

OBSERVATION

① After Deletion of idx 2, all the even & odd elements before it will remain same while all the even will become odd & odd will become even after Selection point.

② So, we can Find Ans without Actual Removal

$$\begin{array}{ccccccc} & & 0 & 1 & 2 & 3 & 4 \\ A[\,] & = & [4, & 1, & 3, & 7, & 10] \end{array}$$

Sum of odd = Sum of odd [0 to 1] + Sum of even [3 to 4]

= 1 + 10

= 11

Quiz 9 :- what will be the Sum of elements at ODD indices in the resulting array after removal of index 3?

$$\begin{array}{cccccccccccc} & & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ A[\,] & = & \{2, & 3, & 1, & 4, & 0, & -1, & 2, & -2, & 10, & 8\} \end{array}$$

Sum of odd = Sum of odd [0 to 2] + Sum of even [4 to 9]

= 15

Quiz 10:- What will be the Sum of elements at EVEN indices in the resulting array after removal of index 3?

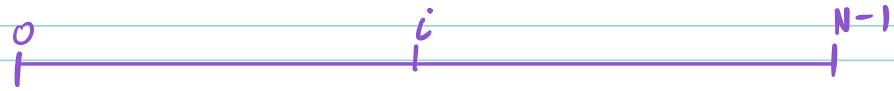
$$\begin{array}{cccccccccccc} & & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ A[\,] & = & [2 & 3 & 1 & 4 & 0 & -1 & 2 & -2 & 10 & 8] \end{array}$$

Sum of even = Sum of even [0 to 2] + Sum of odd [4 to 9]

= 8

OBSERVATION

1.



$$S_o = S_o(0 \text{ to } (i-1)) + S_e((i+1) \text{ to } (N-1))$$

$$S_e = S_e[0 \text{ to } (i-1)) + S_o((i+1) \text{ to } (N-1)]$$

2. How to get the Ans for these Component

$$\boxed{S_o[L-R] = \begin{cases} Pf_o[R] & (L==0) \\ Pf_o[R] - Pf_o[L-1] & (L!=0) \end{cases}}$$

$$\boxed{S_e[L-R] = \begin{cases} Pf_e[R] & (L==0) \\ Pf_e[R] - Pf_e[L-1] & (L!=0) \end{cases}}$$

3. Hence we can update observation 1.

$$S_o = (Pf_o[i-1]) + (Pf_e[N-1] - Pf_e[i])$$

$$S_e = (Pf_e[i-1]) + (Pf_o[N-1] - Pf_o[i])$$

Final Code

```
int Pfe [N];   } we will be creating
int Pfo [N];   } this
int Count = 0;
for (i -> 0 to N-1) {
        Se = 0;   } After deleting i^th index
        So = 0;   }
        So = (Pfo[i-1]) + (Pfe[N-1] - Pfe[i])
        Se = (Pfe[i-1]) + (Pfo[N-1] - Pfo[i])

        if (Se == So) {
                Count ++
        }
}
Print Count;
```

★ In which case the code will fail?

↳ when $i = 0$ then Se & So will give get (-1) as the index.

update the above Code :-

```
if (i == 0) {
    So = (Pfe[N-1] - Pfe[i])
    Se = (Pfo[N-1] - Pfo[i])
}
```

TC ⇒ iterations = 3N

complexity = O(N)

SC ⇒ O(N)

NEXT CLASS

- carry forward.
- subarray

DOUBT SESSION

i	j	iterations
N	[0 N]	N
N/2	[0 N/2]	N/2
N/4	[0 N/4]	N/4
⋮		⋮
1	[0 1]	1

log N terms

$$T.I = N + N/2 + \frac{N}{4} + \frac{N}{8} \ldots\ldots + 1$$

$$\text{Sum of GP} = \frac{a(1-r^n)}{1-r} \quad (\text{if } r<1)$$

$$= \frac{N(1-(1/2)^{\log N})}{1-\frac{1}{2}}$$

$$= 2N\left(1 - \frac{1}{2^{\log N}}\right)$$

$$= 2N - \frac{2N}{2^{\log N}}$$

$$TC = O\left(2N - \cancel{\frac{2N}{2^{\log N}}}\right)$$

$$= O(2N)$$

$$= O(N)$$

$\underline{625}$

2

```
while (i>2):
    i = 25√i
```

$$T.T. = \boxed{N} \rightarrow \sqrt[25]{N} \rightarrow \sqrt[625]{N} \rightarrow \sqrt{\ } \rightarrow \rightarrow 2$$

$O(\log(\log(N)))$