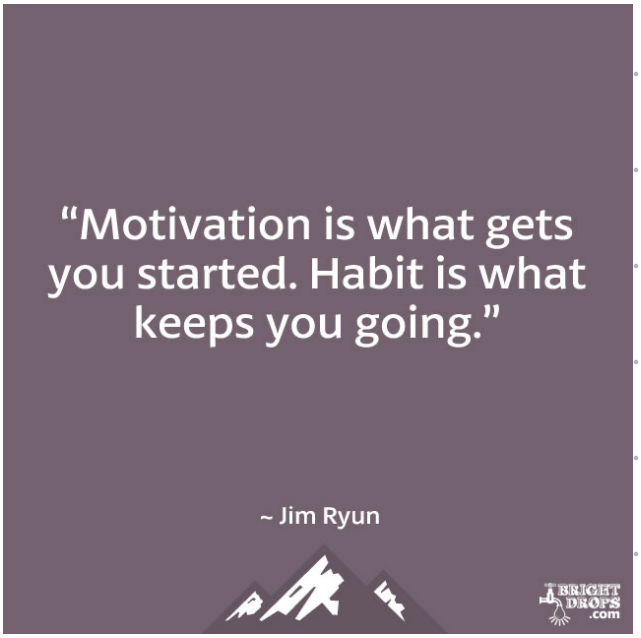
Prefix Sum



Good Morning

Today's content

01. Range sum query
02. Prefix sum construction
03. Sum of even indexed elements
04. Special index

01. Given arr[N] elements & Q queries

For each query : Given L & R calculate & print sum of all elements in range [L R], L & R both has to be calculate

Note :- L & R are array indices such that $0 \le L \le R < N$

Constraints

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

$-11 \le A[i] \le 10^9$

$0 \le L \le R < N$

arr[10] =

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

Q = 5

i =	L	R	
0	4	8	→ 9
1	3	7	→ 10
2	1	3	→ 12
3	0	4	→ 14
4	7	7	→ -9

L = Q[i][0]

R = Q[i][1]

Idea → For each query, we have to iterate from L to R and add all elements ↓

print the sum

```
void rangesum ( int [] A, int [][] Q)
{
    for (i=0 ; i< Q.length ; i++){
        int L = Q[i][0]
        int R = Q[i][1]
        sum = 0;
        for( j=L ; j≤R; j++){
            sum = sum + A[j];
        }
        print (sum);
    }
}
```

TC : O(Q*N)

SC : O(1)

According to the constraint

Constraints

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

$-11 \le A[i] \le 10^9$

$0 \le L \le R < N$

BF Approach = Q * N

$= 10^5 * 10^5 = 10^{10}$ TLE

X

Q: Given Indian Cricked Team score, for first 10 overs of Batting.

After every over, total score is given as:

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

* runs scored in 7th over = Total runs after 7th over − Total runs after 6th over

$$= 65 - 49 \Rightarrow 16$$

* Runs scored from 6th to 10th over =

$$\underbrace{sc[1-10]}_{97} = sc[1-5] + sc[6-10]$$

$$97 = 31 + x$$

$$x = 97 - 31 = 66$$

* Runs scored in 10th over =

$$sc[1-10] = sc[1-9] + \underbrace{sc[10-10]}_{x}$$

$$97 = 88 + x$$

$$x = 97 - 88 = 9$$

* Runs scored from 3rd to 6th over = $sc[6] - sc[3-1]$

$$\Rightarrow 49 - 8 = 41$$

* Runs scored from 4th to 9th over $\Rightarrow sc[9] - sc[4-1]$

$$\Rightarrow 88 - 14$$

$$= 74$$

or[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

cumulative sum array

Q = 5

	L	R
0	4	8
1	3	7
2	1	3
3	0	4
4	7	7

→ psum[8] - psum[4-1] = 18 - 9 = 9

→ psum[7] - psum[3-1] = 15 - 5 = 10

→ psum[3] - psum[1-1] = 9 - (-3) = 12

→ psum[4] = 14

→ psum[7] - psum[7-1] = 15 - 24 = -9

Range sum query

```
if (L != 0)   ans = psum[R] - psum[L-1]
else          ans = psum[R]
```

* Construct psum array

A[] = { 10 32 6 12 20 1 }
0 1 2 3 4 5

psum[] = { 10 42 48 60 80 81 }
0 1 2 3 4 5

psum[i] = sum of all ele from 0 to i

A[] = { 10 32 6 12 20 1 }

0 1 2 3 4 5

psum[] = { 10 42 48 60 80 81 }

0 1 2 3 4 5

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]

```
if (i==0)  pf[0] = A[0]
else { pf[i] = pf[i-1] + A[i] }
```

```
void rangesum (int [] A, int [][] Q)
{
    // construct psum[] array                    n = A.size()

    long [] psum = new long [n]

      psum[0] = A[0]

      for (i=1 ; i<n; i++){

          psum[i] = psum[i-1] + A[i];

      }

    // Answer all queries

    for (i=0 ; i< Q.length ; i++){

          int L = Q[i][0]

          int R = Q[i][1]

          if (L==0) { print (psum[R]) }

          else { print ( psum[R] - psum[L-1]) ;

    }

}
```

TC: O(Q+N)

SC: O(N)

03. Given ar[N] elements & Q queries.

For each query : Given L & R calculate & print sum of even indexed ele in given range [L R]

Note :- L & R are array indices such that $0 \le L \le R < N$

Constraints

$1 \le N, Q \le 10^5$
$1 \le A[i] \le 10^9$
$0 \le L \le R < N$

Eg :- ar[10] :

2	4	3	7	9	8	6	3	4	9
0	1	2	3	4	5	6	7	8	9

Queries ↓

	0	1		Ans
0	4	8	→	19
1	3	9	→	19
2	2	7	→	18
3	0	4	→	14

Brute force → {TODO}

Obs → Odd indexes are going to contribute 0 to the sum.

Eg:- ar[10] :

2	4	3	7	9	8	6	3	4	9
0	1	2	3	4	5	6	7	8	9

pe[10] =

2	2	5	5	14	14	20	20	24	24
0	1	2	3	4	5	6	7	8	9

```
pe[0] = A[0]

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

// Answer all queries

for (i=0 ; i< Q.length ; i++){
    int L = Q[i][0]
    int R = Q[i][1]

    if (L == 0) { print (psum[R]) }

    else { print ( psum[R] - psum[L-1]);

}
```

* Queries on sum of all odd indexed elements

↓

{TODO}

HW →

03. Given ar[N] elements & Q queries.

For each query : Given L & R calculate & print sum of even ele in given range [L R]

Note :- L & R are array indices such that $0 \le L \le R < N$

Constraints

$1 \le N, Q \le 10^5$
$1 \le A[i] \le 10^9$
$0 \le L \le R < N$

ar[10] :

2	4	3	7	9	8	6	3	4	9
0	1	2	3	4	5	6	7	8	9

L = 0
R = 3

sum = 6

L = 3
R = 7

sum = 14

02. Special Index

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

Note → Special indexes are those after removing which sum of all even indexed ele = sum of all odd indexed ele.

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

i	A[i]	Se	So	
0	{3 2 7 6 -2} (0 1 2 3 4)	8	8	Yes
1	{4 2 7 6 -2} (0 1 2 3 4)	9	8	No
2	{4 3 7 6 -2} (0 1 2 3 4)	9	9	Yes
3	{4 3 2 6 -2} (0 1 2 3 4)	4	9	No
4	{4 3 2 7 -2} (0 1 2 3 4)	4	10	No
5	{4 3 2 7 6} (0 1 2 3 4)	12	10	No

count = 2

Quiz 1 $A[] = \{4, 1, 3, 7, 10\}$

indices: 0, 1, 2, 3, 4

remove 2nd idx

$A[] = \{4, 1, 7, 10\}$

indices: 0, 1, 2, 3

$S_o = 1 + 10 = 11$

Quiz 2 $A[] = \{2, 3, 1, 4, 0, -1, 2, -2, 10, 8\}$

indices: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

remove 3rd index

$A[] = \{2, 3, 1, 0, -1, 2, -2, 10, 8\}$

indices: 0, 1, 2, 3, 4, 5, 6, 7, 8

$S_o = 3 + 0 + 2 + 10 = 15$

Quiz 3 $A[] = \{2, 3, 1, 4, 0, -1, 2, -2, 10, 8\}$

indices: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

remove 3rd index

$A[] = \{2, 3, 1, 0, -1, 2, -2, 10, 8\}$

indices: 0, 1, 2, 3, 4, 5, 6, 7, 8

$S_e = 2 + 1 + (-1) + (-2) + 8 \Rightarrow 8$

Sum of even indexed ele after removal of 3rd idx=

Sum of even idx ele from 0 to 2 + sum of odd idx ele from 4 to 9

A[] = { 2 3 1 4 0 -1 2 -2 10 8 }
0 1 2 3 4 5 6 7 8 9

remove 5th index

A[] = { 2 3 1 4 0 2 -2 10 8 }
0 1 2 3 4 5 6 7 8

S_0 = sum of odd idx ele from 0 to 4 + sum of even idx ele from 6 to 9

Remove i idx

S_o = ?

S_e = ?

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

MA[ ] = { 3 4 6 7 2 }
0 1 2 3 4

S_o = 4 + 7 in MA
1 3

S_o = sum of all odd idx ele in A from 0 to i-1 + sum of all even index ele in A from i+1 to n-1

= podd[i-1] + peven[i+1 to n-1]
L R

= peven[R] - peven[L-1]

= peven[n-1] - peven[i+1-1]

S_o = podd[i-1] + peven[n-1] - peven[i]

```
int countspecial (int [] A){

    int peven[n]
    int podd [n]      {TODO}

    count = 0

    for ( i=0 ; i<n ; i++){

        // removing i^th idx

        So = peven[n-1] - peven[i]
        if(i>0){ So = So + podd[i-1] }

        Se = podd[n-1] - podd [i]
        if(i>0){ Se = Se + Peven[i-1] }

        if (So == Se) count ++ ;

    }

    return count

}
```

TC : O(n)

SC : O(n)