Prefix Sum

Content

- Q Queries { sum L – R }
- Q Queries { sum L – R } even index
- Special index.

Amreshwar	M S Haseeb Khan
Anil	Mahesh
ankit shukla	Manohar A N
ANUBHAV RAMNANI	MOHAMMAD ALI
Arunava Basak	Narayan
Arunkumar	Nikhil D
Ashish Kumar	Nutan
Bhavesh Pandey	Pranjul Kesharwani
Bhushan Ganesh Shelar	Prasanna Kumar D R
Deepshikha Arora	Praveen Kumar
Dev Raj Gaur	Priyank Varshney
Dhanashree Sagane	Rahul
Dhasthagiri Reddy	Rishav Ghosh
Dhruv	Sai Kiran JNR
Dipika Malik	Shaik Lal Jan Basha
Farzana kauser	Shambhavi Belligatti
Harikrishnan A	Shivam Shiv
Harshad Marathe	Shivanand Patil
Hemant Verma	Subhash
Induja	Suresh
Iti	Uddeepta Saikia
jeevanantham	Ved Verma
Kanhu	

Real world Example

Stock Portfolio Performance Tracking

Given an array representing daily profit or loss from a particular stock over a period of days.
Write a function that calculate the total profit or loss over a given range of days.

	0	1	2	3	4	5	6	7	8	9
Prices [] =	-5	10	20	40	50	-10	80	-90	-20	-10

Start day	End day	Net profit or loss
0	9	65
1	4	120
0	0	-5
7	9	-120
2	7	90

1	2	3	4
10	20	40	50

Q> Given an integer array with N elements & Q queries.

For each query L, R

— Calculate the sum of all elements from index L to index R

Range Sum Queries

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

Query

L	R	sum(L to R)
4	8	9
3	7	10
1	3	12
0	4	14
7	7	-9

4	5	6	7	8
5	2	8	-9	3

3	4	5	6	7
4	5	2	8	-9

1	2	3
6	2	4

0	1	2	3	4
-3	6	2	4	5

7
-9

Brute force

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

```
for q ⟶ 0 to Q-1 {          ⟶ Q times
    l = L[q]
    r = R[q]
    sum = 0
    for i ⟶ l to r {        ⟶ (r-l+1) times
        sum += A[i]
    }
    print(sum)
}
```

Worst case sum all values N times

TC : O(Q * N)

SC : O(1)

Scoreboard of Cricket after i^{th} over has completed.

0	1	2	3	4	5	6	7	8	9	10
0	2	8	14	29	31	49	65	79	88	97

How many runs were scored in 7^{th} over?

After 7^{th} over end = 65 Runs $65 - 49 = 16$

After 6^{th} over end = 49 Runs

How many runs were scored from 6^{th} to 10^{th} over

0	1	2	3	4	5	6	7	8	9	10
0	2	8	14	29	31	49	65	79	88	97

After 10^{th} $\longrightarrow$ 97 runs

After 5^{th} $\longrightarrow$ 31 runs = 66 runs

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

0	1	2	3	4	5	6	7	8	9	10
0	2	8	14	29	31	49	65	79	88	97

After 10^{th} over $\longrightarrow$ 97 runs

After 9^{th} over $\longrightarrow$ 88 runs $97 - 88 = 9$

How many runs were scored from 3rd to 6th over ?

0	1	2	3	4	5	6	7	8	9	10
0	2	8	14	29	31	49	65	79	88	97

49 - 8 = 41

S[6] - S[2] = 41

49 runs scored [0-6]

- 8 runs scored [0-2

How many runs were scored from 4th to 9th over ?

0	1	2	3	4	5	6	7	8	9	10
0	2	8	14	29	31	49	65	79	88	97

$$S[9] - S[3] = 88 - 14$$
$$= 74$$

→ Runs scored in cricket are provided in a cumulative manner.

→ Lets try to create cumulative sum array

Prefix sum array.

How to create prefix sum array ?

Prefix sum ⟶ Cumulative sum of the array.

$P[i] \longrightarrow$ sum of all elements from 0 to i

$$A = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \end{array}$$

$$\begin{array}{cccccc} & 0 & 1 & 2 & 3 & 4 \\ A = & [\,2 & 5 & -1 & 7 & 1\,] \end{array}$$

$P[0]$ —— 2

$P[1]$ —— $2 + 5 = 7$

$P[2]$ —— $2 + 5 + (-1) = 6$

$P[3]$ —— $2 + 5 + (-1) + 7 = 13$

$P[4]$ —— $2 + 5 + (-1) + 7 + 1 = 14$

$$\begin{array}{cccccc} & 0 & 1 & 2 & 3 & 4 \\ A = & [\,2 & 5 & -1 & 7 & 1\,] \\ P = & [\,2 & 7 & 6 & 13 & 14\,] \end{array}$$

QUIZ

Calculate the prefix sum array for

10	32	6	12	20	1
0	1	2	3	4	5
10	42	48	60	80	81

Code to create prefix sum array ?

$P[i] \longrightarrow$ sum of all elements from 0 to i

Bruteforce $\longrightarrow$ Do as the question says
Most straightforward approach

```
for i ⟶ 0 to N-1 {
    psum = 0
    for j ⟶ 0 to i {
        psum += A[j]
    }
    print(psum)
}
```

TC : $O(N^2)$

$A = 1, 2, 3$

i

1
3
6

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \end{array}$$
$$A = [2, 5, -1, 7, 1]$$

$P[0] = A[0]$

$P[1] = A[0] + A[1]$

$P[2] = A[0] + A[1] + A[2]$

$P[3] = A[0] + A[1] + A[2] + A[3]$

$P[4] = A[0] + A[1] + A[2] + A[3] + A[4]$

$P[0] = A[0]$

$P[1] = P[0] + A[1]$

$P[2] = P[1] + A[2]$

$P[3] = P[2] + A[3]$

$P[4] = P[3] + A[4]$

$P[i] = P[i-1] + A[i]$

if $i > 0$

if $i == 0 \longrightarrow A[0]$

Optimised

```
P[N]    // initialize prefix sum
P[0] = A[0]
for i → 1 to N-1 {
    P[i] = P[i-1] + A[i]
}
```

TC: $O(N)$

SC: $O(N)$

$$
\begin{array}{cccccccc}
 & & & 0 & 1 & 2 & 3 & 4 \\
A & = & [& 2 & 5 & -1 & 7 & 1\ ] \\
P & = & [& 2 & 7 & 6 & 13 & 14\ ] \\
 & & & 0 & 1 & 2 & 3 & 4
\end{array}
$$

$P[r]$

$\underbrace{A_0 \quad A_1 \ldots A_{l-1}}_{P[l-1]} \ldots A_l \ldots\ldots\ldots\ldots A_r \ldots\ldots A_{N-1}$

sum from (l to r)

$= P[r] - P[l-1]$

if $l == 0$ $\quad P[r]$

How to answer the queries?

$$\begin{array}{ccccccccccccc} & & & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ A & = & [& -3 & 6 & 2 & 4 & 5 & 2 & 8 & -9 & 3 & 1 \;] \\ P & = & & -3 & 3 & 5 & 9 & 14 & 16 & 24 & 15 & 18 & 19 \end{array}$$

Query

L	R	sum(L to R)	
4	8	P[8] − P[3]	18 − 9 = 9
3	7	P[7] − P[2]	15 − 5 = 10
1	3	P[3] − P[0]	12
0	4	P[4]	14
7	7	P[7] − P[6]	15 − 24 = −9

sum[L, R] = P[R] − P[L−1]

if L == 0   P[R]

Optimised

[3 6 2 4 5 2 8 9 3 1]

```
// Create Prefix sum array
P[N]    // initialize prefix sum
P[0] = A[0]
for i → 1 to N-1 {
    P[i] = P[i-1] + A[i]
}
```

TC: O(N)

SC: O(N)

```
for q ⟶ 0 to Q-1 {          ⟶ Q times
    l = L[q]
    r = R[q]
    sum = 0
    if (l==0) {
        sum = P[r]
    } else {
        sum = P[r] - P[l-1]
    }
    print(sum)
}
```

$A = [1\ 2\ 3\ 4]$

$L = [0, 1, 1, 3, 2, 1]$

$R = [0, 2, 3, 3, 2, 3]$

TC : $O(N+Q)$

SC : $O(N)$

$A = [1\ \cancel{2}^{3}\ \cancel{3}^{6}\ \cancel{4}^{10}]$

$P = [1\ 3\ 6\ 10]$

Can we improve space complexity ?

yes we the given A[] as prefix sum array.

SC : $O(1)$

22:35

~~A[0] = A[0]~~

```
for i ⟶ 1 to N-1 {
    A[i] = A[i-1] + A[i]
}
```

TC

Given an array of size N and Q queries with start L and end R index.

Print the sum of all even indexed elements from L-R

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

Query L	R	even indexed sum(L to R)
1	3	1
2	5	5
0	4	7
3	3	0

1	2	3
3	1	6

Bruteforce   for every query iterate from $l \longrightarrow r$
if $(i \% 2 == 0) \longrightarrow$ add to sum
else ignore

	0	1	2	3	4	5
A =	2	3 → 0	1	6 → 0	4	5 → 0
P =	2	2	3	3	7	7

// Create Prefix sum array
P[N]   // initialize prefix sum
P[0] = A[0]
for $i \longrightarrow$ 1 to N-1 {

```
val = 0
if (i%2 == 0) { // i at even index
    val = A[i]
}
P[i] = P[i-1] + val
}
```

val = 0

$$\begin{array}{cccccccc} & & 0 & 1 & 2 & 3 & 4 & 5 \\ A & = & [2 & 3 & 1 & 6 & 4 & 5] \\ P & = & 2 & 2 & 3 & 3 & 7 & 7 \end{array}$$

Quiz ⟶

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

$$\begin{array}{cccccccc} & & 0 & 1 & 2 & 3 & 4 & 5 \\ A & = & [2 & \cancel{3}0 & 1 & \cancel{6}0 & 4 & \cancel{5}0] \\ P & = & [2 & 2 & 3 & 3 & 7 & 7] \end{array}$$

sum (L to R) = P[R] − P[L−1]

Query L	R	even indexed sum (L to R)	
1	3	P[3] − P[0]	= 3−2 = 1
2	5	P[5] − P[1]	= 7 − 2 = 5
0	4	P[4]	= 7
3	3	P[3] − P[2]	= 3−3 = 0

Pseudocode

```
// calculate Psum
for q ⟶ 0 to Q-1 {        ⟶ Q times
    l = L[q]
    r = R[q]
    sum = 0
    if (l==0) {
        sum = P[r]
    } else {
        sum = P[r] - P[l-1]
    }
    print(sum)
}
```

TC: O(N+Q)

Given an array of size N and Q queries with start L and end R index.

Print the sum of all odd indexed elements from L-R

$$A = \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$
$$A = [\,2 \quad 3 \quad 1 \quad 6 \quad 4 \quad 5\,]$$

```
// Create Prefix sum array
P[N]    // initialize prefix sum
P[0] = 0
for i ⟶ 1 to N-1 {
    val = 0
    if (i%2 == 1) {   // i at odd index
        val = A[i]
    }
    P[i] = P[i-1] + val
}
```

Special Index

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

Special indices are those after removing them

sum of all odd indexed elements == sum of all even indexed elements

NOTE : after removal of index, indices will change

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

i	A (indices 0 1 2 3 4)	S_O == S_E	cnt
0	3 2 7 6 -2	8 == 8	+1
1	4 2 7 6 -2	8 != 9	
2	4 3 7 6 -2	9 == 9	+1
3	4 3 2 6 -2	9 != 4	
4	4 3 2 7 -2	10 != 4	
5	4 3 2 7 6	10 != 12	

cnt = 2

$A =$	4	1	~~3~~	7	10
	0	1	~~2~~	3	4

remove $i = 2$

4	1	7	10
0	1	2	3

sum of odd idx = 11

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

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

remove $i = 3$

sum of odd idx = 15

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

sum of all even idx = 8

Observation

remove $i = 3$

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

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

sum of all odd idx

sum of all odd idx = sum of all odd idx (0 − 2)
+ sum of all even idx (4 − 9)

PO ⟶ Prefix sum for all odd idx
PE ⟶ Prefix sum for all even idx
= PO[2] + PE[9] − PE[3]

---

sum of all even idx = sum of all even idx (0 − 2)
+ sum of all odd idx (4 − 9)

$$A = \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 2 & 3 & 1 & \cancel{4} & 0 & -1 & 2 & -2 & 10 & 8 \end{array}$$

$$\begin{array}{ccccccccc} 2 & 3 & 1 & 0 & -1 & 2 & -2 & 10 & 8 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array}$$

sum of all odd after removing i^{th} idx
= sum of odd (0 , i − 1) + sum of even (i+1 , N−1)
PO[i−1] + PE[N−1] − PE[i]

sum of all even after removing i^{th} idx
= sum of even (0 , i−1) + sum of all odd (i+1 , N−1)
= PE[i−1] + PO[N−1] − PO[i]

Pseudocode

```
// PS for odd idx          PO[N]
// PS for even idx         PE[N]
   se ⟶ sum of even idx
   so ⟶ sum of odd idx
   cnt = 0
for i ⟶ 0 to N-1 {
      if (i == 0) {
           so  =  PE[N-1] - PE[i]
           se  =  PO[N-1] - PO[i]
      } else {
           so  =  PO[i-1] + PE[N-1] - PE[i]
           se  =  PE[i-1] + PO[N-1] - PO[i]
      }

      if (se == so) {
            cnt += 1
      }
}

print (cnt)
```

TC : O(N)

SC : O(N)

0 1 2 3

1 2 3 ~~4~~

$PO[2]$ $\cancel{PE[3] - PE[3]}$ 0

$PO[i-1] + PE[N-1] - PE[i]$

$PE[i-1] + PO[N-1] - PO[i]$

$PE[2] + 0$

$(0 - i-1)$ $(i+1, N-1)$

i

0	1	~~2~~	3	4
10	14	~~9~~	7	6

take even idx from right

10	14	7	6
0	1	2	3

take odd from left

sum of odd idx = 20