#BBM

Amazon

Q Given an array of size N.

& Q queries of the format s & e (s<=e)

s: start index, e: end index

Return the sum of elements from index s to e

A : $\overset{0}{-3}, \overset{1}{6}, \overset{2}{2}, \overset{3}{4}, \overset{4}{5}, \overset{5}{2}, \overset{6}{8}, \overset{7}{-9}, \overset{8}{3}, \overset{9}{1}$

Q : 4

s	e	
1	3	⟶ 12
2	7	⟶ 12
4	8	⟶ 9
0	2	⟶ 5

```
for (i=0; i<Q; i++) {  ⟶ Q iterations
    // scan s & e
    sum = 0;
    for (j=s; j<=e; j++) {   [s, e] → e-s+1 iterations
        sum = sum + A[j];        N iterations
    }
    Print [sum].
}
```

TC : O(QN)

Given the scores of last 10 overs of a match

41	42	43	44	45	46	47	48	49	50
288	312	330	349	360	383	394	406	436	439

Runs scored in 42nd over ⟶ R[42] - R[41]
312 - 288
⇒ 24

Runs scored in 49th over ⟶ R[49] - R[48]
436 - 406
⇒ 30

Runs scored in last 5 overs ⟶ R[50] - R[45]
439 - 360
⇒ 79

Runs scored from 42nd to 45th ⟶ R[45] - R[41]
360 - 288
⇒ 72

	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

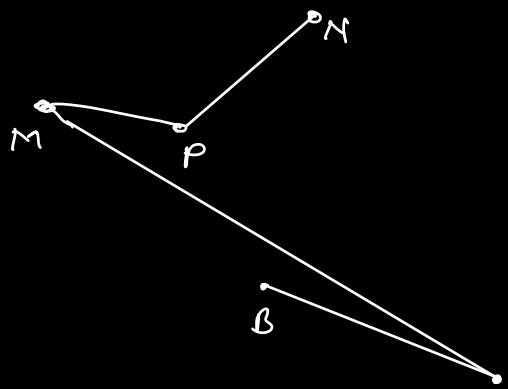
Prefix Sum Array — Every index stores the sum of all elements from start (0) till that index

PS[i] ⇒ sum of elements from index 0 to i

Wanderlust

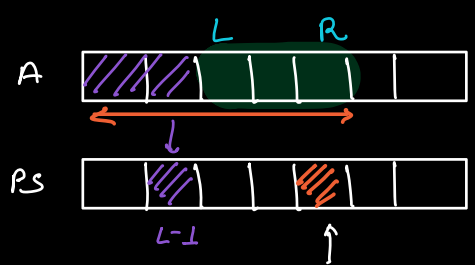
1700

B → 0
C → 250
M → 1300
P → 1400
N → 1700



M → N ⇒ 1700 − 1300 ⇒ 400 km
C → M ⇒ 1300 − 250 ⇒ 1050 km

Sum of elements from index L to R ⇒ PS[R] − PS[L−1]



A : −3, 6, 2, 4, 5, 2, 8, −9, 3, 1 (indices 0 1 2 3 4 5 6 7 8 9)

PS : −3, 3, 5, 9

PS[0] = A[0]

PS[1] = A[0] + A[1] ⇒ PS[0] + A[1]

PS[2] = A[0] + A[1] + A[2] ⇒ PS[1] + A[2]

PS[3] = A[0] + A[1] + A[2] + A[3] ⇒ PS[2] + A[3]

PS[4] = A[0] + A[1] + A[2] + A[3] + A[4] ⇒ PS[3] + A[4]

⋮

PS[i] = A[0] + A[1] + A[2] + ... + A[i−1] + A[i] ⇒ PS[i−1] + A[i]

PS ⟶ new array of size N

PS[0] = A[0];

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

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

}

TC : O(N)

SC : O(N)

Step I → Build the PS — O(N)

Step II → Run a loop Q times & answer all Q ques — O(Q)

TC : O(N+Q)

Range Sum Queries

Direct i

Q Given an array. Return true if there exists an equilibrium index in the array.

EI ⇒ index for which,
Sum of all elements on left side = Sum of all elements on right side

A : 0 1, 1 2, 2 3, 3 4, 4 8, 5 10 → True

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

Brute Force

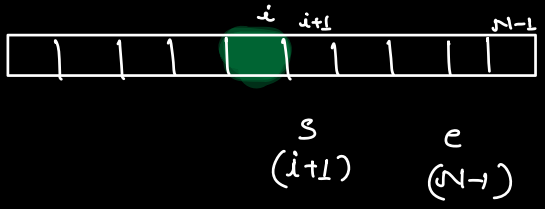
for every index i

Check if $\sum [0, (i-1)] == \sum [(i+1), (N-1)]$

return true,

```
for (i=0; i<N; i++) {                    → O(N)
    SL = 0;   → 0 - (i-1)
    SR = 0;   → (i+1) - (N-1)
    for (j=0; j<i; j++) {
        SL = SL + A[j];
    }
    for (j=i+1; j<N; j++) {              } (N-1)
        SR = SR + A[j];
    }
    if (SL == SR) return true;
}
```

TC : $O(N^2)$
SC : $O(1)$
Extra



PS → Build a prefix sum

```
for ( i=0; i < N; i++) { → O(N)
    SL = PS[i-1], // if i→0, SL→0
    SR = PS[N-1] - PS[i];

    if (SL == SR) ret true;
}
```

$\Sigma[s,e] = PS[e] - PS[s-1]$

iterations $N_{PS} + N_{EI} \Rightarrow 2N$

TC : O(N)

SC : O(N)

Q PayTM Odd

Given an array of size N & Q queries (s, e) for every query return Count of even elements in the index range from s to e.

A : $\overset{0}{-3}, \overset{1}{6}, \overset{2}{2}, \overset{3}{4}, \overset{4}{5}, \overset{5}{2}, \overset{6}{8}, \overset{7}{-9}, \overset{8}{3}, \overset{9}{1}$

Q : 4

s	e	
1	3	$\longrightarrow$ 3
2	7	$\longrightarrow$ 4
4	8	$\longrightarrow$ 2
0	2	$\longrightarrow$ 2

Brute force

for every query
iterate from s to e
& count the even no.

$O(QN)$

	0	1	2	3	4	5	6	7	8	9
A :	-3,	6,	2,	4,	5,	2,	8,	-9,	3,	1
Even array :	0,	1,	1,	1,	0,	1,	1,	0,	0,	0

Now this becomes Q1. $\longrightarrow O(Q+N)$

Amortized : Gradually write off the initial cost of an asset over a period of time

40/-