Q: Given an Array of size N & Q queries of format s & e. Return the sum of array elements from s to e

Amazon

s = Start, e = End

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

Q: 4

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

⇒ **Idea 1: Brute force**

```
for( i= 1 ; i <= Q; i++ ) {
    //s,e given
    sum = 0
    for (j= s; j <= e ; j++ ){
        sum += A[j];
    }
    print(sum);
```

j ∈ [s, e]

e - s + 1

⇒ N iteration in Worst Case

TC: O(Q·N)
SC: O(1)

En Given the score 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

Quiz-1 Runs scored in last 5 overs.

[46, 50]

439 - 360

⇒ 79

Quiz-2 Runs scored in 50th over.

R[50] - R[49]

439 - 436 = 3

Quiz-3 Runs scored in 49th over.

R[49] - R[48]

436 - 406 ⇒ 30

Quiz-4 Runs scored from 42nd to 45th over :-

[42, 45]

R[45] - R[41]

360 - 288

72

100 → 31*

Create a new Array of **Prefix Sum**

⇒ Every index stores the sum of all elements from start (0th index) till that index

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

PS[N-1] → Sum of Array

PS[0] : A[0]

PS[1] : Sum [0-1]

: PS[0] + A[1]

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

PS[3] : PS[2] + A[3]

PS[4] : PS[3] + A[4]

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

i = 0 , PS[0] = A[0]

```
int PS[N]
PS[0] = A[0]
for(i=1; i<N; i++){
    PS[i] = PS[i-1] + A[i]
}
```

TC: O(N)

SC: O(N)

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

S	e	ans
1	3	PS[3] - PS[0] = 9 - (-3) = 12
2	7	PS[7] - PS[2-1] = 15 - 3 = 12
4	8	PS[8] - PS[3] = 18 - 9 = 9
0	2	PS[2] = 5 ↳ Sum[0-2]

Sum[s-e] : PS[e] - PS[s-1]

s == 0 , Sum[0-e] : PS[e]

TC: O(N + Q) → O(N+Q)

N → Build PS

Q → Queries

SC: O(N)

↓

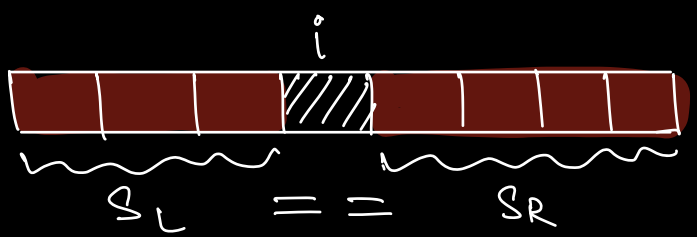
PS Array

Q. Equilibrium Index

Given an Array, return true if there exists an equilibrium index in the Array.

Equilibrium Index:

Sum of elements on left side = Sum of elements on right side



A: 1, 2, 3, 4, 8, 10 (indices 0 1 2 3 4 5)

$S_L = 10$ ⇑ EQ $S_R = 10$

$S_L == S_R \Rightarrow$ EQ ✓

Quiz

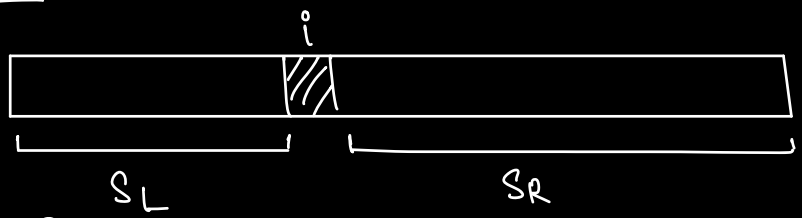
A: -7, 1, 5, 2, -4, 3, 0 (indices 0 1 2 3 4 5 6)

$S_L = -1$ ↑ $S_R = -1$

$S_R = 0$, $S_L = 0$ EQ ✓

→ True

Brute force



for every index i { if ($S_L == S_R$) return true; }

S_L : Sum [0, i-1]

S_R : Sum [i+1, N-1]

for every index i :-

(i == 0, S_L = 0)

(i == N-1, S_R = 0)

{ find S_L : [0, i-1]
find S_R : [i+1, N-1] } O(N)

if ($S_L == S_R$)
return true;

}

TC: O(N·N) ⇒ $O(N^2)$

SC: O(1)

#

S_L = Sum of elements from $[0, i-1]$

= $PS[i-1]$

S_R = Sum of elements from $[i+1, N-1]$ (s, e)

= $PS[N-1] - PS[i]$

Steps

1) Build PS array $\Rightarrow O(N)$

2)
```
SL = PS[i-1]
SR = PS[N-1] - PS[i]
if (SL == SR)
    return true
```
} for every index i. $O(N)$

TC: $O(N)$

SC: $O(N)$

10:39 pm

Special Index x