Arrays : Prefix Sum

✓ 12th Aug – Friday

13th – Saturday

14th – Sunday

✓ 15th Aug – Monday

Agenda

- Prefix Sum Introduction
- Problems based on Prefix Sum

Q1 Given N array elements & Q queries on same array.
For each query calculate sum of all elements in given range - [L, R]
Note: L & R are indices such that L <= R

arr[10] =	[-3	6	2	4	5	2	8	-9	3	1]
	0	1	2	3	4	5	6	7	8	9

Q = 6

L	R		
4	8	:	9
3	7	:	10
1	3	:	12
0	4	:	14
6	9	:	3
7	7	:	-9

```
solve ( int []arr ) {
    Q = input ()
    while  Q > 0 {
        Q - = 1
        l, r = input()
        sum from index
          l to r
        for  i = l; i <= r; i++
            ≡
        print (sum)
    }
}
```

Q iteration

O(N) →

Constraints

$Q \leq 10^5$

$N \leq 10^5$

Time - O(Q N)

Space - O(1)

Q2

Given Indian Cricket Team scores 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
Scores :	2	8	14	29	31	49	65	79	88	97

Cumulative sum

score[10] - score[9]

Total runs scored in last over: 97 - 88 = 9 runs

Total runs scored in 7th over: Quiz 1

score[7] - score[6]

65 - 49 = 16 runs

Total runs scored in overs 6th to 10th: Quiz 2

score[10] - score[6]

[~~1,2,3,4,5,6~~,7,8,9,10] - [~~1,2,3,4,5,6~~] = [7,8,9,10]

score[10] - score[5]

[~~1,2,3,4~~,5,6,7,8,9,10] - [~~1,2,3,4,5~~] = [6,7,8,9,10]

97 - 31 = 66

Total runs scored in overs 3rd to 6th:

score[6] - score[2]

49 - 8 = 41 runs

Total runs from i^{th} over to j^{th} over

score[j] - score[i-1]

Prefix Sum Array

Left to Right

Suffix Sum Array

Right to Left

$$\text{Sum}[i \quad j] = pf[j] - pf[i-1]$$

arr[10] =	[-3	6	2	4	5	2	8	-9	3	1]
	0	1	2	3	4	5	6	7	8	9
pf[10] =	-3	3	5	9	14	16	24	15	18	19

$$pf[i] = sum[0 \quad i]$$

[4, 8] : $pf[8] - pf[3]$

$= 18 - 9$

$= 9$

$i \quad j$

[3, 7] : Quiz 3

$pf[7] - pf[3-1]$

$= pf[7] - pf[2]$

$= 15 - 5$

$= 10$

[i, j] :
(i <= j)

$$\begin{cases} \text{if } i > 0 - pf[j] - pf[i-1] \\ \text{if } i == 0 - pf[j] \end{cases}$$

$[\overset{i}{0}, 3]$:

$$pf[3] - pf[0-1]$$

$$= pf[3] - pf[-1] \rightarrow$$ wrong value / Error

$$\rightarrow pf[3]$$

$$= 9$$

How to construct prefix array ?

N = size of arr

arr[10]	=	[-3	6	2	4	5	2	8	-9	3	1]
		0	1	2	3	4	5	6	7	8	9
pf[10]	=	-3	3	5	9	14	16	24	15	18	19

arr ← input

```
pf [N]
pf [0] = arr [0]

for (i=1 ; i<N ; i+=1)   {
        pf [i] = pf [i-1] + arr [i]

}
```

$pf[i] = sum[0 \quad i]$

$pf[0] = arr[0]$

$pf[1] = arr[0] + arr[1]$

$\Rightarrow pf[1] = pf[0] + arr[1]$

$pf[2] = arr[0] + arr[1] + arr[2]$

$= pf[1] + arr[2]$

$pf[3] = arr[0] + arr[1] + arr[2] + arr[3]$

$= pf[2] + arr[3]$

$pf[i] = pf[i-1] + arr[i]$

Code for Q1

```
solve ( int []arr ) {
    // Construct prefix sum
    Q = input()
    while (Q > 0){
        Q -= 1
        l, r = input()
        sum from index l to r = sum[l  r]
        if l > 0 :
            ans = pf[r] - pf[l-1]
        else
            ans = pf[r]
        print (ans)
    }
}
```

O(N) time

sum[i j]

$\begin{cases} \text{if } i > 0 - pf[j] - pf[i-1] \\ \text{if } i == 0 - pf[j] \end{cases}$

Q iterations

Time = O(N + Q)

Space = O(N)

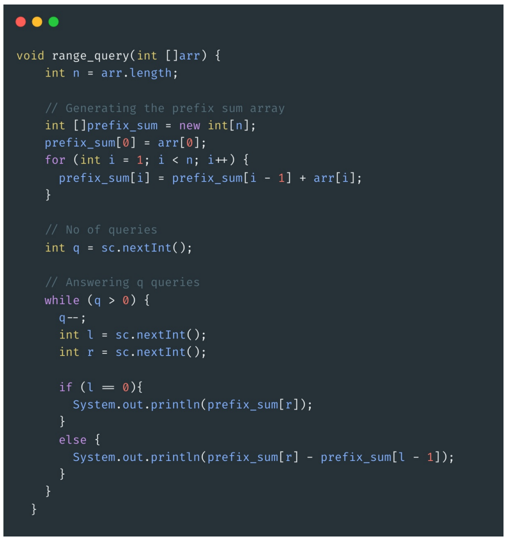
Constraints

$N \leq 10^5$

$Q \leq 10^5$

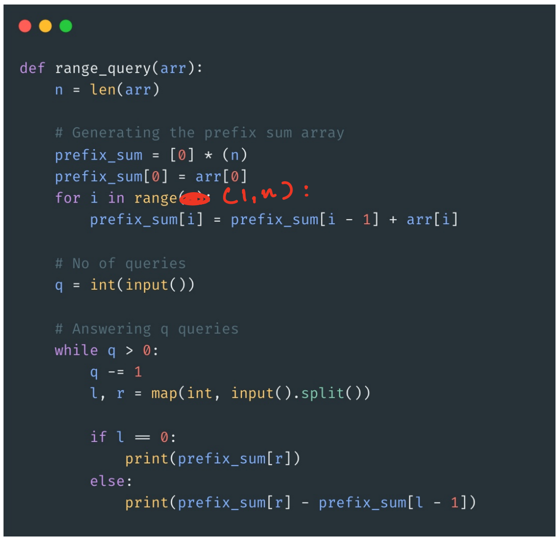
Earlier we had $O(N * Q)$

Java



```java
void range_query(int []arr) {
    int n = arr.length;

    // Generating the prefix sum array
    int []prefix_sum = new int[n];
    prefix_sum[0] = arr[0];
    for (int i = 1; i < n; i++) {
      prefix_sum[i] = prefix_sum[i - 1] + arr[i];
    }

    // No of queries
    int q = sc.nextInt();

    // Answering q queries
    while (q > 0) {
      q--;
      int l = sc.nextInt();
      int r = sc.nextInt();

      if (l == 0){
        System.out.println(prefix_sum[r]);
      }
      else {
        System.out.println(prefix_sum[r] - prefix_sum[l - 1]);
      }
    }
  }
```

Python



```python
def range_query(arr):
    n = len(arr)

    # Generating the prefix sum array
    prefix_sum = [0] * (n)
    prefix_sum[0] = arr[0]
    for i in range(1,n):
        prefix_sum[i] = prefix_sum[i - 1] + arr[i]

    # No of queries
    q = int(input())

    # Answering q queries
    while q > 0:
        q -= 1
        l, r = map(int, input().split())

        if l == 0:
            print(prefix_sum[r])
        else:
            print(prefix_sum[r] - prefix_sum[l - 1])
```

Generally we follow above approach.

Modify the original array

arr[10] = [	-3	6	2	4	5	2	8	-9	3	1]
	0	1	2	3	4	5	6	7	8	9
new arr =	-3	3	5	9	14	16	24	15	18	19

```
for (i=1; i<N; i++) {
    arr[i] = arr[i] + arr[i-1]
}
```

⇒ **Benefit**

No extra space

Space – O(1)

Drawback

Lost the original data

Adobe amazon hike

Q2: Equilibrium Index

Given N array elements, count no of equilibrium index.

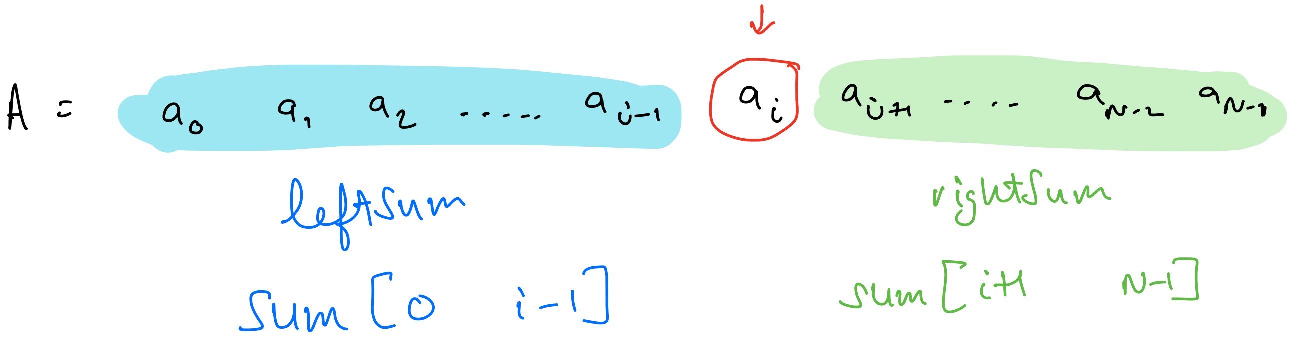
An index i is said to be equilibrium index if:

Sum of all elements left of i^{th} index = Sum of all elements right of i^{th} index

Sum[0, i-1] Sum[i+1, N-1]

Note:

- if i == 0, leftSum = 0
- If i == N-1, rightSum = 0



$A = a_0 \quad a_1 \quad a_2 \quad \dots \quad a_{i-1}$
a_i
$a_{i+1} \quad \dots \quad a_{N-2} \quad a_{N-1}$
leftSum
Sum [0 i-1]
rightSum
sum [i+1 N-1]

Example

		0	1	2	3
arr [4]	=	-3	2	4	-1
left	:	0	-3	-1	3
right	:	5	3	-1	0

Equilibrium Index = 2

Count = 1

Quiz 4

Options : 2, [✓ 3], 4, 5

Example

		0	1	2	3	4	5	6
arr [7]	:	-7	1	5	2	-4	3	0
left	:	0	-7	-6	-1	1	-3	0
right	:	7	6	1	-1	3	0	0

Equilibrium Index = 3, 6

Count = 2

Example

		0	1	2
ar[3]	=	3	-2	2
left	:	0	3	1
right	:	0	2	0

Count= 1

Example

		0	1	2	3	4	5	6
ar[7]	=	3	-1	2	-1	1	2	1
left	:							
right	:							

Logic & Pseudocode

→ for every index if i is equilibrium or not.

1) Run a loop – O(N)

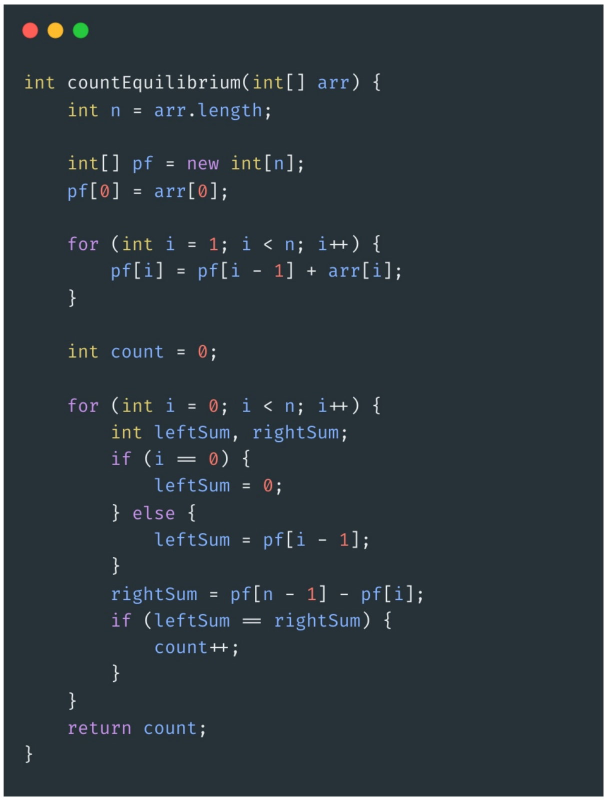
for i in range (N):

2) Optimise ??

```
solve ( int [] arr ] {
    count = 0
    for ( i=0; i<N; i+=1 ) {
        // check if i is equilibrium
        leftSum = sum [0   i-1]   ← pf [i-1]
        rightSum = sum [i+1   N-1]   ← pf [N-1] - pf [i]
        if ( leftSum == rightSum)
            count += 1
    }
    return count
}
```

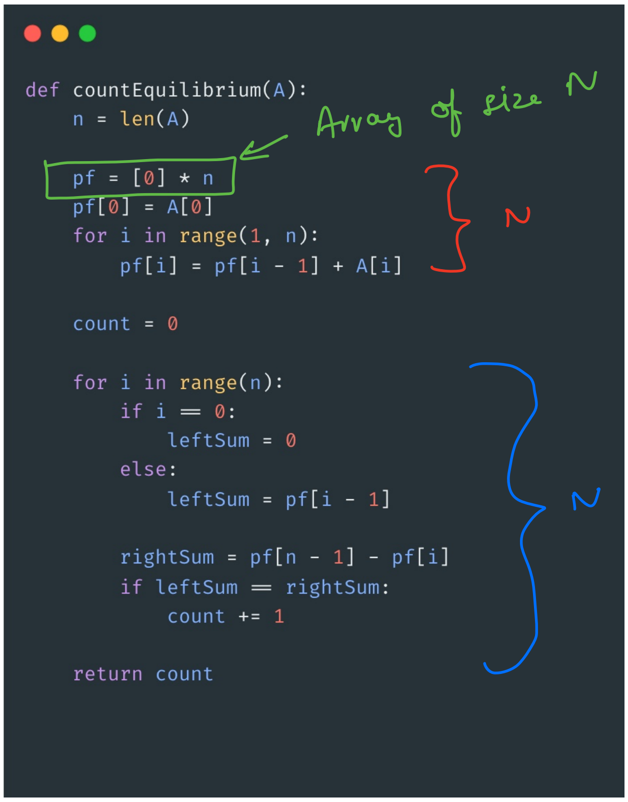
$$\text{sum}[l \;\; r] \begin{cases} \text{if } l>0 - pf[r] - pf[l-1] \\ \text{if } l==0 - pf[r] \end{cases}$$

Java



```java
int countEquilibrium(int[] arr) {
    int n = arr.length;

    int[] pf = new int[n];
    pf[0] = arr[0];

    for (int i = 1; i < n; i++) {
        pf[i] = pf[i - 1] + arr[i];
    }

    int count = 0;

    for (int i = 0; i < n; i++) {
        int leftSum, rightSum;
        if (i == 0) {
            leftSum = 0;
        } else {
            leftSum = pf[i - 1];
        }
        rightSum = pf[n - 1] - pf[i];
        if (leftSum == rightSum) {
            count++;
        }
    }
    return count;
}
```

Python



```python
def countEquilibrium(A):
    n = len(A)

    pf = [0] * n
    pf[0] = A[0]
    for i in range(1, n):
        pf[i] = pf[i - 1] + A[i]

    count = 0

    for i in range(n):
        if i == 0:
            leftSum = 0
        else:
            leftSum = pf[i - 1]

        rightSum = pf[n - 1] - pf[i]
        if leftSum == rightSum:
            count += 1

    return count
```

Total = N + N = 2N

Time - O(N)

Space - O(N)

Q3

Given N array elements, construct a pfeven[N] such that,

Pfeven[i] = Sum of all even indices from [0 - i]

indices	:	0	1	2	3	4	5
arr[6]	:	3	-2	4	6	-3	5
				+		+	
pfeven[6]	:	3 →	3	7 →	7	4 →	4

For even index

pfeven[i] = arr[i] + pfeven[i-1]

For odd index

pfeven[i] = pfeven[i-1]

Quiz 5

indices	:	0	1	2	3	4
arr[5]	:	2	4	3	1	5
pfeven[5]	:	2 →	2	5 →	5	10

pfeven[i] = sum[0 i] of even index

pfeven[3] = sum of all even indexed elements till i=3

= arr[0] + arr[2]

Pseudocode

```
                              size = N
Pfeven [N]

pfeven [0] = arr[0]

for (i=1; i<n; i++) {                        i%2 == 1
        if( i is odd) {
            pfeven [i] = pfeven [i-1]
        }
        else {
            pfeven[i] = arr[i] + pfeven [i-1]
        }
}
```

Time - O(N)

Space - O(N)

Q4 Given N array elements & Q queries, for each query calculate sum of all even indices in given range.

	0	1	2	3	4	5	6	7
ar[8] :	3	4	-2	8	6	2	1	3
pfeven[8] :								

Queries

[2 6] :

[3 7] :

$[i \quad j]$:

TODO:

Generate pfodd[] to compute sum of all odd indices in given range [i j]

pfodd[i] = sum of all odd indexed elements till i.

Code - Try on your own

- 2-3 lines change

Say we delete 4th index. How will the array look like ?

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

Q5 Special Index

HARD

An index is said to be special index, if after deleting it

Sum of all even index = Sum of all odd index

Count how many special index are there ?

Doubts

Thank You

App = Notability

Next Session — Prefix Sum 2

Q4 and Q5

Good Night

Thank You

Monday