30/6/2022

Prefix Sum

Today's quote:- If at first you don't succeed
TRY TRY TRY AGAIN

Today's content

1. Prefix Sum
2. Problems of Prefix Sum.

Q1 Given N → no of elements array elements & Q → no of queries. For each query - calculate sum of all elements in range [L, R].

Note :- L & R are indices such that L <= R

$1 <= N, Q <= 10^5$

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

Queries

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

idea → For every query, calculate the sum [iterate from L to R]

TC: O(Q*N)
SC: O(1)

[TLE]

Pseudo code

```
void fun (arr, N) {
    Q <- take i/p [no of queries]

    while (Q > 0) {
        Q--;

        l, r <- take i/p

        sum = 0
        for (i = l; i <= r; i++) {
            sum += arr[i];
        }
        print (sum);
    }
}
```

Q =

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

SB: { 2 8 14 29 31 49 65 79 88 97 }

Cumulative Sum.

Total runs scored in the last over = 97 − 88 = 9

score[10] − score[9]

Total runs scored in 7th over = 65 − 49 = 16

score[7] − score[6]

Total runs scored in 6th to 10th over = 97 − 31 = 66

score[10] − score[6−1]

Total runs scored in overs 3rd to 6th = 49 − 8 = 41

score[6] − score[3−1]

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

= score[j] − score[i−1]

idea: store cumulative sum / prefix sum

arr[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

Queries

L	R	Ans
4	8	→ psum[8] - psum[4-1] ⇒ 18 - 9 = 9
3	7	→ psum[7] - psum[3-1] = 15 - 5 = 10
1	3	→ psum[3] - psum[1-1] = 9 - (-3) = 12
0	4	→ psum[4] ⇒ 14
7	7	→ psum[7] - psum[7-1] ⇒ 15 - (24) = -9

[l r → psum[r] - psum[l-1]]

if l == 0, psum[r]

[psum(i) = sum of all the elements from index 0 to i]

How to calculate prefix array.

{ 3 -2 4 5 6 }

psum[0] = arr[0] = 3

psum[1] = arr[0] + arr[1]

psum[1] = psum[0] + arr[1]

psum[2] = arr[0] + arr[1] + arr[2]

psum[2] = psum[1] + arr[2]

psum[3] = arr[0] + arr[1] + arr[2] + arr[3]

psum[3] = psum[2] + arr[3]

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

Pseudo code

```
psum[0] = arr[0]

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

TC: O(N)
SC: O(N)

Pseudo code for Q1

```
void fun ( arr , N ) {

    psum[N];
    psum[0] = arr[0];
    for i -> 1 to N-1 {
        psum[i] = psum[i-1] + arr[i]
    }

    q <- no of queries.

    while (q > 0) {
        q--;
        l, r <- take i/p.

        if (l == 0) {
            print (psum[r]);
        }
        else {
            print (psum[r] - psum[l-1]);
        }
    }
}
```

N

Q

TC: O(N+Q)

SC: O(N)

Can we modify the i/p array?

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

```
for i -> 1 to N-1 {
    arr[i] = arr[i-1] + arr[i]
}
```

Advantage: SC is optimized

Disadvantage: initial elements will be lost.

Q { Amazon, Adobe, Hike } Equilibrium index

Given N array elements, count no of equilibrium index.

An index i is said to be EI if:

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

Note: if $i == 0$, leftsum = 0

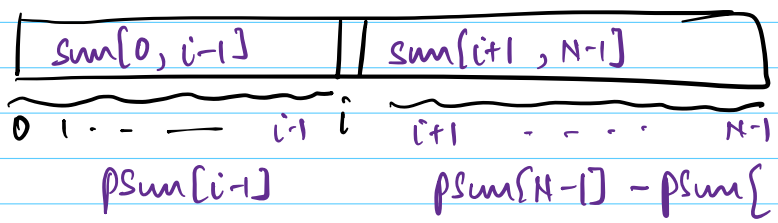
if $i = N-1$, rightsum = 0

Eg arr → { -3 2 4 -1 }
0 1 2 3

leftsum	0	-3	-1	3	ans = 1
Rightsum	5	3	-1	0	

arr → [-7 1 5 2 -4 3 0]
0 1 2 3 4 5 6

LS →	0	-7	-6	-1	1	-3	0
RS →	7	6	1	-1	3	0	0



$psum[L, R] \Rightarrow psum[R] - psum[L-1]$

$psum[i+1, N-1] \Rightarrow psum(N-1] - psum[i+1-1]$

$= psum[N-1] - psum[i]$.

logic & pseudo code

Use prefix sum.

{ psum[i-1] == psum[N-1] - psum[i] }

check this for every index.

```
int countEl (arr, N) {

    psum[N]
    psum[0] = arr[0]
    for i -> 1 to N-1 {
        psum[i] = psum[i-1] + arr[i]
    }                                   } N

    count = 0

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

        if (psum[i-1] == psum[N-1] - psum[i])
            count++;

    }                                   } N
    return count;
}
```

TC: O(N)

SC: O(N) / O(1)

TODO when i = 0

Meet at 10:36 pm.

Q Given N array elements & Q queries.
For each query l to r. Find count of even numbers in given range.

Eg arr[10] : { 2 4 3 7 9 8 6 3 4 9 }
0 1 2 3 4 5 6 7 8 9

Queries

L	R
4	8
3	9
0	4

Bruteforce : For every query, iterate from L to R & find count of even nos.

TC: O(Q*N)
SC: O(1)

```
void fun (arr, N) {
    q <= i/p
    while (q>0) {
        q--;
        l, r <= i/p
        count = 0;
        for (i = l; i<= r; i++) {
            if (arr[i] % 2 == 0)
                count++;
        }
        print (count);
    }
}
```

Optimization

arr[10] : { 2 4 3 7 9 8 6 3 4 9 }
0 1 2 3 4 5 6 7 8 9

even → 1
odd → 0

arr[10] : { 1 1 0 0 0 1 1 0 1 0 }
0 1 2 3 4 5 6 7 8 9

psum : { 1 2 2 2 2 3 4 4 5 5 }
0 1 2 3 4 5 6 7 8 9

{ count of even nos from l to r = psum[r] - psum[l-1] }

l = 0, r = 4 → psum[4] - psum[0-1] → 0

```
void fun (arr, N) {
    for i -> 0 to N-1 {
        if (arr(i) % 2 == 0)
            arr[i] = 1
        else
            arr[i] = 0
    }

    for i -> 1 to N-1 {
        arr[i] += arr[i-1]
    }
}
```

```
arr[0] = (arr[0] % 2 == 0) ? 1 : 0;

for i -> 1 to N-1 {
    arr[i] = (arr[i] % 2 == 0) ? 1 : 0;

    arr[i] += arr[i-1];
}
```

```
q ⇐ take i/p
while (q>0){
   q--
   l, r ⇐ i/p                    TC: O(N+Q)
   if(l==0) print arr[r]         SC: O(1)
   else print (arr[r] - arr[l-1]);
}
}
```

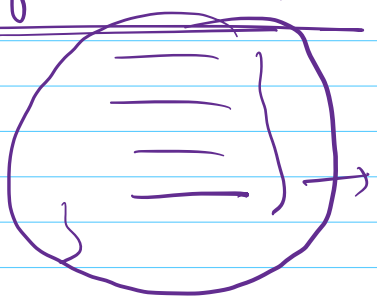
Rounds

1 GHz

10^9 instructions / sec.

for - $i \to 0$ to 10^6 {

N



$10^6 - 10^7 - 10^8$

1 sec $\to 10^9$ inst

$\Rightarrow 10 \times 10^8$ instn

N

arr[N][N] O(N²)

O(lgN)

O(lgN)