

Agenda

- 1) Prefix sum intro
- 2) questions based on prefix sum
- 3) carry forward
- 4) ag pairs
- 5) Subarray intro

Q. Given $A[]$ and Q queries (every query has L & R). Find sum of elements of Array from L to R .

$A[] = [-3 \ 6 \ 2 \ 4 \ 5 \ 2 \ 8 \ -9 \ 3 \ 1]$
 0 1 2 3 4 5 6 7 8 9

constraints

$1 \leq n \leq 10^5$

$1 \leq q \leq 10^5$

L	R	Sum
0	3	9
1	5	19
4	4	5
5	8	4

	0	1	
0	0	3	→ 9
1	1	5	→ 19
2	4	4	→ 5
3	5	8	→ 4

Q

Brute force

```
void solve (int[] A, int[][] Q) {
```

```
    for (int i=0; i<Q.length; i++) {
```

```
        int L = Q[i][0];
```

```
        int R = Q[i][1];
```

```
        int sum = 0;
```

```
        for (int j=L; j<=R; j++) {
```

```
            |    sum += A[j];
```

```
            |
```

```
            |    sum;
```

```
        }
```

$A = [6 \ -3 \ 1 \ 4 \ 2]$
 0 1 2 3 4

	0	1	
0	1	3	= 2
1	2	4	= 7

Q

$i=0, \ L=1 \ R=3, \ \text{sum}=2$

$i=1, \ L=2 \ R=4, \ \text{sum}=7$

constraints

$1 \leq n \leq 10^5$

$1 \leq q \leq 10^5$

TC: $O(q \cdot n) \rightarrow \text{TLE}$

SC: $O(1)$

Optimised Idea

Given the scores of the 10 overs of a cricket match

2, 8, 14, 29, 31, 49, 65, 79, 88, 97

How many runs were scored in just 7th over?

$$\begin{aligned} 7^{\text{th}} \text{ Over} &= 65 - 49 \\ &= 16 \end{aligned}$$

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

How many runs were scored from 6th to 10th over(both included)?

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

$$\begin{aligned} &\text{runs till } 10^{\text{th}} \text{ over} - \\ &\text{runs till } 5^{\text{th}} \text{ over} \\ &= 97 - 31 = 66 \end{aligned}$$

How many runs were scored in just 10th over?

$$97 - 88 = 9$$

How many runs were scored from 3rd to 6th over(both included)?

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

$$\begin{aligned} &\text{runs scored till } 6^{\text{th}} - \\ &\text{run scored till } 2^{\text{nd}} \\ &= 49 - 8 = 41 \end{aligned}$$

Intro to prefix sum

$$A[] = [-3 \quad 6 \quad 2 \quad 4 \quad 5 \quad 2 \quad 8 \quad -9 \quad 3 \quad 1]$$

0 1 2 3 4 5 6 7 8 9

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

$$PS[] = [-3 \quad 3 \quad 5 \quad 9 \quad 14 \quad 16 \quad 24 \quad 15 \quad 18 \quad 19]$$

0 1 2 3 4 5 6 7 8 9



$PS[i]$ = sum of elements from 0 to i or
sum of elements till i^{th} index

$$A[] = [-3 \quad 6 \quad 2 \quad 4 \quad 5 \quad 2 \quad 8 \quad -9 \quad 3 \quad 1]$$

0 1 2 3 4 5 6 7 8 9

$$PS[] = [-3 \quad 3 \quad 5 \quad 9 \quad 14 \quad 16 \quad 24 \quad 15 \quad 18 \quad 19]$$

0 1 2 3 4 5 6 7 8 9

L	R	Sum
1	5	$PS[5] - PS[0] = 16 - (-3) = 19$
5	8	$PS[8] - PS[4] = 18 - 14 = 4$
0	3	$PS[3] = 9$
4	4	$PS[4] - PS[3] = 14 - 9 = 5$

if $(L == 0)$ {

Sum = $PS[R]$;

}

else {

Sum = $PS[R] - PS[L-1]$;

}

Q. Given an $A[]$ and Q queries with L, R . For every query, return sum of all even indexed elements from L to R .

$A[] = [2 \ 3 \ 1 \ 6 \ 4 \ 5]$
 0 1 2 3 4 5

L	R	sum
0	4	7
1	5	5
2	4	5
2	2	1

1) Brute force :

TC: $O(q \times n)$

SC: $O(1)$

2) Optimised logic

$A[] = [2 \ 3 \ 1 \ 6 \ 4 \ 5]$
 0 1 2 3 4 5

$PS[] = [2 \ 2 \ 3 \ 3 \ 7 \ 7]$
 0 1 2 3 4 5

↑
 $PS[i] \Rightarrow$ sum of even indexed values from 0 to i

if $(i \% 2 \neq 0)$ {

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

}

else {

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

}

L R sum

0 4 $PS[4] = 7$

1 5 $PS[5] - PS[0] = 5$

2 4 $PS[4] - PS[1] = 5$

2 2 $PS[2] - PS[1] = 1$

```
void fun (int [] A, int [][] Q) {
```

```
    // construct prefix sum array
```

```
    int [] ps = new int [A.length];
```

```
    ps[0] = A[0];
```

```
    for (int i = 1; i < ps.length; i++) {
```

```
        if (i-1 < 0) {
```

```
            ps[i] = ps[i-1];
```

```
        }
```

```
        else {
```

```
            ps[i] = ps[i-1] + A[i];
```

```
        }
```

```
    }
```

```
    // find the ans for each query
```

```
    for (int i = 0; i < Q.length; i++) {
```

```
        int L = Q[i][0];
```

```
        int R = Q[i][1];
```

```
        int sum = 0;
```

```
        if (L == 0) {
```

```
            sum = ps[R];
```

```
        }
```

```
        else {
```

```
            sum = ps[R] - ps[L-1];
```

```
        }
```

```
        sopLn(sum);
```

```
    }
```

```
}
```

TC: $O(n+q)$

SC: $O(n)$

↳ creation of
ps[]

a. Given a string s of lowercase characters, return the count of pairs (i, j) such that $i < j$ and $s[i]$ is 'a' and $s[j]$ is 'g'.

$s = a b e g a g$
0 1 2 3 4 5

ans = 3

$s = a g a b g g a$
0 1 2 3 4 5 6

ans = 5

$s = a c g d g a g$
0 1 2 3 4 5 6

ans = 4

$s = b c a g g a a g$
0 1 2 3 4 5 6 7

ans = 5

const. $1 \leq n \leq 10^7$

Find count of pairs in which 'a' is before 'g'

```
int countAgPair (string str) {
```

```
    int n = str.length();
```

```
    int ans = 0;
```

TC: $O(n^2)$ TLE

```
    for (int i = 0; i < n; i++) {
```

```
        if (str.charAt(i) == 'a') {
```

```
            int temp = 0;
```

```
            for (int j = i + 1; j < n; j++) {
```

```
                if (str.charAt(j) == 'g') {
```

```
                    temp++;
```

```
                }
```

```
            }
```

```
            ans += temp;
```

```
        }
```

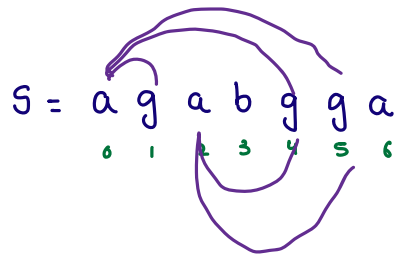
```
    }
```

```
    return ans;
```

```
}
```

SC: $O(1)$

→ Optimised



$$cg = 0 + 2 + 3$$

$$ans = 0 + 2 + 3$$

```
int countagpairs (string str) {
```

```
    int n = str.length();
```

```
    int cg = 0, ans = 0;
```

```
    for (int i = n-1; i >= 0; i--) {
```

```
        if (str.charAt(i) == 'g') {
```

```
            cg++;
```

```
        }
```

```
        else if (str.charAt(i) == 'a') {
```

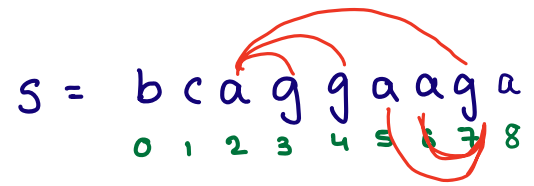
```
            ans += cg;
```

```
        }
```

```
    }
```

```
    return ans;
```

```
}
```



$$cg = 0 + 0 + 1 + 1 + 3$$

$$ans = 0 + 0 + 1 + 1 + 3 = 5$$

Intro to Subarrays

↳ continuous part of an Array

i) continuous part of an Array

ii) single element is also a subarray & complete array is also a subarray

iii) Order matters

$A = \{ 2 \quad 3 \quad 8 \quad 5 \quad 7 \quad 10 \}$

valid subarray or not?

$\{ 2 \quad 3 \quad 5 \quad 10 \}$ X

$\{ 3 \quad 8 \quad 5 \quad 7 \}$ ✓

$\{ 7 \}$ ✓

$\{ 3 \quad 8 \quad 7 \quad 5 \}$ X

$A[] = \{2, 4, 1, 6, -3, 7, 8, 4\}$

Which of the following is a valid subarray?

20 users have participated

- | | | |
|-----|--------------------|-----|
| A | $\{1, 6, 8\}$ ✗ | 5% |
| B | $\{1, 4\}$ ✗ | 0% |
| C | $\{6, 1, 4, 2\}$ ✗ | 0% |
| ✓ D | $\{7, 8, 4\}$ ✓ | 95% |

$A = \{ \underset{0}{2} \quad \underset{1}{3} \quad \underset{2}{8} \quad \underset{3}{5} \quad \underset{4}{7} \quad \underset{5}{10} \}$ s, e

$1, 4 \rightarrow \{ 3 \quad 8 \quad 5 \quad 7 \}$

$2, 2 \rightarrow \{ 8 \}$