



Array - Carry forward & Subarrays

TABLE OF CONTENTS

1. Count 'a-g' pairs
2. Sub-arrays
3. Print a Subarray
4. Print all Subarrays
5. Min-Max



Contest Information

Date : 15th November

Time : 7 am IST

Duration : 1.5 hrs

Post contest there will be contest discussion class on same day from 8:30 am IST.

Passing marks : 75%

Total 4 questions in contest from the topics taught in class

How to prepare for contest?

Solve all assignments and keep PSP >90%

Question

Calculate the prefix sum array for following array:-

10 32 6 12 20 1

Choices

- [] [10,42,48,60,80,81]
- [] [10,42,49,60,79,81]
- [] [42,48,60,80,81,10]
- [] [15,43,58,61,70,82]

Question

What is the generalised equation to find the sum of elements from L to R ?

Choices

- [] $\text{sum}[L \ R] = \text{pf}[R] - \text{pf}[L-1]$
- [] $\text{sum}[L \ R] = \text{pf}[R] - \text{pf}[L]$
- [] $\text{sum}[L \ R] = \text{pf}[R-1] - \text{pf}[L]$
- [] $\text{sum}[L \ R] = \text{pf}[R-1] - \text{pf}[L-1]$

Question

What is the time taken to find sum of element from index L to R, assuming you have computed prefix sums array ?

Choices

- [] $O(N)$
- [] $O(1)$
- [] $O(\log(N))$
- [] $O(N^2)$



Count 'a-g' pairs

< Question > : 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'.





BF Idea

Consider all the pairs & check if $s[i] == 'a'$ & $s[j] == 'g'$.

```
int CountPairs (String s) {
```

```
    Count = 0;
```

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

```
        if (s[i] == 'a') {
```

```
            for (j=i+1 ; j < s.length ; j++) {
```

```
                if (s[j] == 'g')
```

```
                    count++;
```

```
}
```

```
y
```

```
y
```

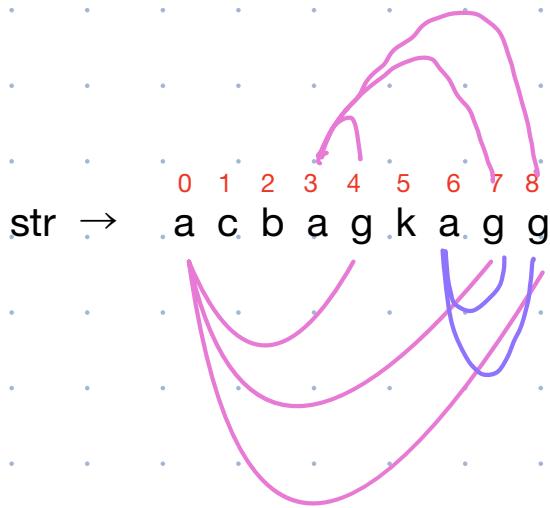
```
return count;
```

```
y
```

T.C : $O(N^2)$

S.C : $O(1)$

$1 \leq N \leq 10^6$

**Idea**

	0	1	2	3	4	5	6	7	8
a	a	c	b	a	g	k	a	g	g
Count of 'a'	1	1	1	2	2	2	3	3	3
ans	0	0	0	0	2	2	2	5	8

ans : 8

```
int CountPairs(String s) {  
    count_a = 0 ;  
    ans = 0 ;  
  
    for(i=0; i < s.length; i++) {  
        if ( s[i] == 'a' ) {  
            count_a++ ;  
        }  
        else if ( s[i] == 'g' ) {  
            ans += count_a ;  
        }  
    }  
    return ans ;  
}
```

T.C : O(N)
S.C : O(1)



Subarrays

- contiguous part of an array.
 - formed by selecting a range of elements from the array.
 - can have one or more elements and must be a contiguous part of the original array.
-
- Single element of an array is a subarray.
 - Whole array is a subarray.
 - Empty array is not a subarray.

0	1	2	3	4	5	6	7	8
4	1	2	3	-1	6	9	8	12

Which of the following is a subarray?

- [2, 3, -1, 6] ✓
- [9] ✓
- [4, 1, 2, 3, -1, 6, 9, 8, 12] ✓
- [4, 12] ✗
- [1, 2, 6] ✗
- [3, 2, 1, 4] ✗



Example: $\text{arr}[:] \rightarrow [2\ 4\ 1\ 6\ -3\ 7\ 8\ 4]$

Which of the following is a subarray?

- a. $[1, 6, 8]$ X
- b. $[1, 4]$ X
- c. $[6, 1, 4, 2]$ X
- d. $[7, 8, 4]$ ✓

Representation of a subarray

- By specifying **Start** and **End** index of the subarray
- By specifying **Start_{Index}** and **length** of the subarray

0	1	2	3	4	5	6	7	8
4	1	2	3	-1	6	9	8	12

$$l_i = 3$$

$$e_i = 6$$

$$S_i = 3$$

$$l = 4$$



QUIZ

- How many subarrays of the following array start from index 0?

0	1	2	3	4	5	6
4	2	10	3	12	-2	15

[4]

[4, 2]

[4, 2, 10]

[4, 2, 10, 3]

[4, 2, 10, 3, 12]

[4, 2, 10, 3, 12, -2]

[4, 2, 10, 3, 12, -2, 15]

ans : 7

QUIZ

- How many subarrays of the following array start from index 1?

0	1	2	3	4	5	6
4	2	10	3	12	-2	15

[2]

[2, 10]

[2, 10, 3]

[2, 10, 3, 12]

[2, 10, 3, 12, -2]

[2, 10, 3, 12, -2, 15]

ans : 6



Total number of subarrays

0	1	2	3
4	2	10	3

$$N = 4$$

s	e
0	0
0	1
0	2
0	3

s	e
1	1
1	2
1	3

s	e
2	2
2	3

s	e
3 • • • • • •	3 • • • • •

Total No. of Subarrays :

4
2

3
N-1

α
 $N - \alpha$

1
2-3

No. of Subarrays starting from Index 0 : N

No. of Subarrays starting from Index 1 : N - 1

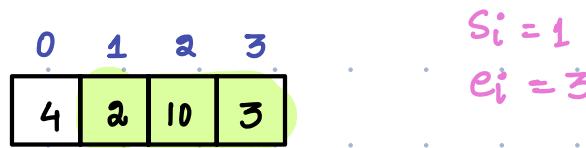
No. of Subarrays starting from Index 2 : N - 2

No. of Subarrays starting from Index (N-1) : 1

$$\begin{aligned} \text{Total No of Subarrays} &= N + (N-1) + (N-2) \dots 2+1 \\ &= \frac{N(N+1)}{2} \end{aligned}$$



< Question > : Given an array, start index (s_i) and end index (e_i). Print subarray from s_i to e_i .



```
void PrintSubarray (int []arr, int si, int ei) {
```

```
    for(i = si ; i <= ei ; i++)  
        print (arr[i]);
```

T.C : O(N)
S.C : O(1)



<Question>: Print all the possible sub-arrays of the given array.

0 1 2 3
[5, 7, 3, 2]

[5]
[5 7]
[5 7 3]
[5 7 3 2]

[7]
[7 3]
[7 3 2]

[3]
[3 2]

[2]

$$\frac{N(N+1)}{2} = \frac{4 \times 5}{2} = 10$$

void PrintSubarray (int [] arr, int n) {

 || Generate all Subarrays.

 for (int si = 0 ; si < arr.length ; si++) {

 for (int ei = si ; ei < arr.length ; ei++) {

 || Print the Subarray :

 for (i = si ; i <= ei ; i++) {

 |
 y print (arr[i]);

y

y

y

T.C. : O(N³)
S.C. : O(1)

[5 7 3 2]

$s_i : 0 \rightarrow 3$ 0 1

$e_i : s_i \rightarrow 3$ 2

$i :$ 0 1

5

5 7

5 7 3

5 7 3 2

7

7 3

|

|



Min Max

< Question > : Given an array of N integers, return the length of smallest subarray which contains both maximum and minimum elements of the array.

$1 \leq N \leq 10^6$

0 1 2 3 4 5 6 7 8 9
arr[] → [1 2 3 1 3 4 6 4 6 3]

Min Element : 1

Ans : 4

Max Element : 6

QUIZ

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

Min Element : 1

Ans : 3

Max Element : 6

0 1 2 3 4 5
arr[] → [8 8 8 8 8 8]

Min Element : 8

Ans : 1

Max Element : 8



Idea Consider all the Subarrays $\left(\frac{N(N+1)}{2}\right)$. Check if a subarray which has both MAX & MIN. & length is MINIMUM.

T.C: $O(N^3)$

$(10^6)^3 \rightarrow \text{TLE}$

1 4 1 5 2 3 6



Observation

1. There must be exactly one occurrence of min & max element.

2. Min and max elements should be the end point of subarray.

case-1: [min - - - max] closest index OF MIN

case-2: [max - - - min] closest index OF MAX



Idea

0	1	2	3	4	5	6	7	8	9	10
2	2	6	4	5	1	5	2	6	4	1

last_min_index : -1

last_max_index : -1

ans : INT-MAX

Min-value : 1

Max-value : 6

i	A[i]	last_min_index	last_max_index	ans
0	2	-1	-1	INT-MAX
1	2	-1	-1	INT-MAX
2	6	-1	2	INT-MAX
3	4	-1	2	INT-MAX
4	5	-1	2	INT-MAX
5	1	5	2	$5-2+1 = 4$
6	5	5	2	4
7	2	5	2	4
8	6	5	8	$8-5+1 = 4$
9	4	5	8	4
10	1	10	8	$10-8+1 = 3$

ans : 3

```
int MinSubarray( int[ ] arr, int n ) {
```

// Find MIN & MAX value of array.
MinValue MaxValue.

last_min_index = -1 ;

last_max_index = -1 ;

ans = INT_MAX ;

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

```
    if( arr[i] == minValue ) {
```

 last_min_index = i ;

```
        if( last_max_index != -1 ) {
```

 ans = Math.min(ans, last_min_index - last_max_index + 1) ;

y

```
    else if( arr[i] == maxValue ) {
```

 last_max_index = i ;

```
        if( last_min_index != -1 ) {
```

 ans = Math.min(ans, last_max_index - last_min_index + 1) ;

y

 return ans ;

T.C : O(N)

S.C : O(1)

To solve this problem, we can generate all possible subarrays of the array using two nested loops.