Amazon

Q1 Given a string of lowercase char, return the count of pairs (i, j) st

indexes.

$i < j$

$s[i] = 'a'$

$s[j] = 'g'$

0	1	2	3	4	5	6
a	c	g	d	g	a	g

Pairs ⇒ (0,2) (0,4) (0,6) (5,6)

ans = 4

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

Pairs ⇒ (2,3) (2,4) (2,7) (5,7) (6,7)

ans = 5

Brute: Use 2 for loops for i<j

TC: $O(n^2)$

Obs: ..... g .....

'g' forms a pair with every 'a' on the left

	0	1	2	3	4	5	6	7
Ex	a	c	b	a	g	k	a	g
count_a = 0	1	1	1	2	2	2	3	3
ans = 0	0	0	0	0	2	2	2	5

Code

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

TC: $O(n)$

SC: $O(1)$

Subarray: Continuous part of an array

1) Complete array is subarray of itself
2) Single element of array is also subarray
3) Empty subarray is NOT valid.

	0	1	2	3	4	5	6
A:	3	4	5	6	-2	8	10

3, 5, 6 ✗

-2 ✓

8, 10 ✓

8, 10, 3 ✗

⇒ Subarray can be defined by

start end

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

s e

0 0 1 2 3 4 5 6

1 1 2 3 4 5 6

Total no of subarrays =

Subarray starting at 0

+ Subarray starting at 1

+ 2

+ Subarray starting at n-1

0	0	1	1	...	n-1 n-1
0	1	1	2		
0	2	1	3		
⋮		⋮			
0	n-1	1	n-1		
n		n-1		n-2	1

$$n + (n-1) + (n-2) + \dots + 1$$

$$1 + 2 + 3 + 4 + \dots + n$$

$$= \frac{n(n+1)}{2}$$

Q Print all values of a subarray.

```
void printSubarray (int A[], int s, int e){
    for ( i=s ; i<=e ; i++ ){
        print a[i]
    }
}
```

TC: $O(n)$

Q Print all possible subarrays.

```
for ( start = 0 ; start <n ; start ++){
    for ( end = start ; end <n ; end ++){
        printSubarray ( arr, start, end )
    }
}
```

TC: $O(n^3)$

$\underbrace{N + N + N \quad - - - \quad + N}_{N^2 \text{ times}}$

Q3 Given an array, Return the length of smallest subarray which contains

↓ continuous part of array

both MIN & MAX of the array.

	0	1	2	3
A:	100	200	300	400

100 , 200

200, 300, 400

100, 200, 300, 400

100, 200, 400 ✗

400

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

max = 6 min = 1

ans = 4

Quiz

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

max = 6 min = 1

ans = 3

Obs 1 : In the required answer subarray, MAX & MIN should occur exactly once

Max ---- Min ---- Max

Obs 2 : In the required answer subarray,

---- Max ---- Min ----

→ Max & Min will be on the boundaries

- Solutions : MAX ------ MIN OR MIN ------ MAX

- For every MIN, find the closest MAX on the left side
- For every MAX, find the closest MIN on the left side

prev_max_idx: Index of the nearest max on the left

prev_min_idx: Index of the nearest min on the left

	0	1	2	3	4	5
Eg	6	4	1	2	5	6
prev_min_idx = -1	-1	-1	2	2	2	2
prev_max_idx = -1	0	0	0	0	0	5
ans =			3			

For each iteration,

- if a[i] == MAX, ans = min(ans, ?)
  where is the prev MIN prev_min_idx
  what is current pos i
  thus range = [prev_min_idx, i]
  ans = min(ans, i - prev_min_idx + 1)

- Similarly, if a[i] == MIN
  ans = min(ans, i - prev_max_idx + 1)

Pseudocode

1) Find MAX & MIN

2) prev_max = -1 prev_min = -1

ans = n

```
for(i=0; i<n; i++){
    if (a[i] == MAX){
        prev_max_idx = i
    }
    if (a[i] == MIN){
        prev_min_idx = i
    }
    if ( prev_max_idx == -1 ||
         prev_min_idx == -1 )
        continue
    else {
        if (a[i] == MAX)
            ans = min (ans, i - prev_min_idx + 1)
        else if (a[i] == MIN)
```

ans = min (ans, i - prev_max_idx + 1)

}

return ans

TC: O(n)

SC: O(1)