Q Amazon Given a string of lowercase alphabets. Return count of (i, j) such that

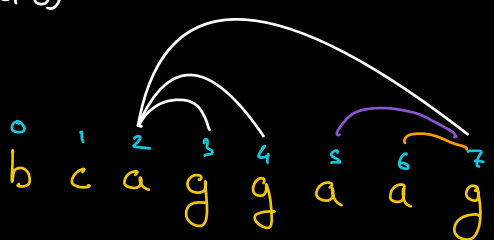
$i < j$
$s[i] \longrightarrow$ 'a'
$s[j] \longrightarrow$ 'g'
} 'ag'

S : a b e g a g

(0, 3) (4, 5)
(0, 5) ⟶ 3

Quiz

(0, 2) (5, 6)
(0, 4)
(0, 6) ⟶ 4



Quiz

(2, 3) (5, 7) (6, 7)
(2, 4)
(2, 7)

```
ans = 0;
for( i=0; i<N; i++){
    if (a[i]=='a'){
        for (j=i+1; j<N; j++){
            if( a[j] == 'g'){
                ans++;
            }
        }
    }
}
```

TC : $O(N^2)$

SC : $O(1)$

a c b a g k a g g

Every 'g' makes a pair with all the 'a' on the left side of it.

	a	c	b	a	g	k	a	g	g
Count a	1	1	1	2	2	2	3	3	3
ans	0	0	0	0	2	2	2	5	8

```
ans = 0;
Count_a = 0;
for (i = 0; i < N; i++) {
    if (S[i] == 'a') {
        Count_a++;
    }
    else if (S[i] == 'g') {
        ans = ans + Count_a;
    }
}
return ans;
}
```

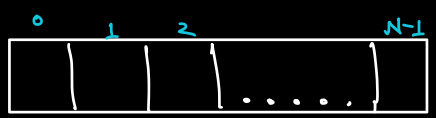
TC : O(N)
SC : O(1)

Amazon
Zeta

Q Given an array. Return the length of smallest sub-array which contains both, the max & min of the array.

Sub-array any contiguous part of an array

How many sub-arrays in an array of size N.



Count of sub-arrays starting from index 0 ⟶ N
Count of sub-arrays starting from index 1 ⟶ N-1
Count of sub-arrays starting from index 2 ⟶ N-2
:
Count of sub-arrays starting from index N-1 ⟶ 1

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

$\Rightarrow \frac{N(N+1)}{2}$

$\approx O(N^2)$

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

Array Max ⟶ 6

Array Min ⟶ 1

[3, 6] ⟶ 4

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

Array Max ⟶ 6

Array Min ⟶ 1

[8, 10] ⟶ 3

- The ans sub-array must have exactly one min & one max.

1 | 1 3 6 | 4 3 2 5

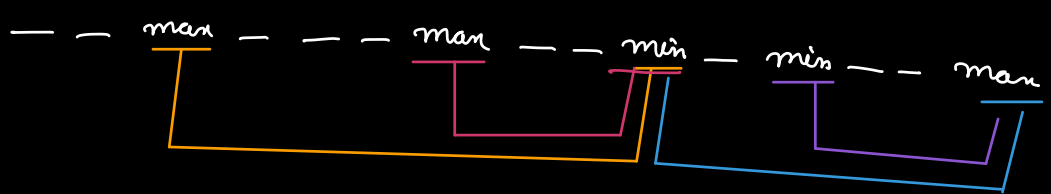
1 | 1 6

- The min & max in the ans sub array will be present in the corner,

6, 1, 2, 3, 4, 5

1, 3, 6, 4, 1

- [max, min]
  Or
  [min, max]



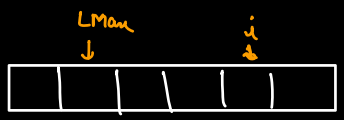
// Find the min of array ⟶ A_{min}

// Find the max of array ⟶ A_{max}

lastMinIndex = -1

lastMaxIndex = -1

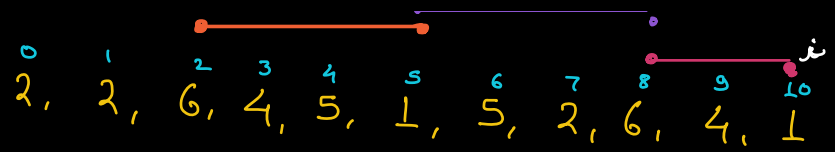
ans = N




```
for (i=0; i<N; i++) {
    if (A[i] == Amin) {
        lastMinIndex = i;
        if (lastMaxIndex >= 0) {
            ans = min(ans, i - lastMaxIndex + 1);
        }
    }
    else if (A[i] == Amax) {
        lastMaxIndex = i;
        if (lastMinIndex >= 0) {
            ans = min(ans, i - lastMinIndex + 1);
        }
    }
}
```

TC : O(N)

SC : O(1)



Lmin : ~~1~~ ~~5~~ 10

Lmax : ~~-1~~ ~~2~~ 8

ans : ~~1~~ ~~4~~ 3

7, 7, 7, 7, 7

Lmin : ~~-1~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

Lmax : -1

ans : 5

max → 7
min → 7

Q Given an array. Count the no. of leaders.

Leader : An element that is greater than all the elements on the left side.

A[i] > [0, i-1]

2, 5, 3, 4, 17, 16

[2, 5, 17] ⟶ 3

- For every element A[i] in the array
  - iterate from 0 to i-1 & check if A[i] is greater than all elements in this range

TC : O(N²)

7, 3, 2, 9, 12, 6, 8, 20

max : ~~7~~ ~~9~~ ~~12~~ 20

ans : ~~1~~ ~~2~~ ~~3~~ 4