Q1) Given a char [], calculate no. of pairs (i,j) such that $i < j$ && $s[i] = $ 'a' && $s[j] = $ 'g'.

$$s[8] = \left\{ \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ b & a & a & g & d & c & a & g \end{array} \right\}$$

5 pairs

i	j
1	3
2	3
1	7
6	7
2	7

3 2

4 7

$i < j$ ×

$s[i] ==$ 'd'

$$s[8] = \left\{ \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ b & c & a & g & g & a & a & g \\ \times & \times & +3 & \times & \times & +1 & +1 & \times \end{array} \right\}$$

Ans : 5

$$s[7] = \left\{ \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \boxed{a} & c & g & d & g & a & g \end{array} \right\}$$

Ans : 4

$i<j$ $s[i] = \text{'a'}$ $s[j] = \text{'g'}$

0	1	2	3	4	5	6	7
a	c	g	a	d	g	a	g
+3			+2			+1	

Ans : 6

TC: $O(N^2)$

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

Problem: For every a, we are repeatedly iterating the same array from L-R to count total no. of gs

$\boxed{\text{No. of gs on the right of every a}}$

s[8] = { b c a g g a a g }

0	1	2	3	4	5	6	7
b	c	a	g	g	a	a	g
		agc += cg	cg++	cg++	agc += cg	agc += cg	cg++
		agc = 5	cg = 3	cg = 2	agc = 2	agc = 1	cg = 1

cg = 0   ag count = 0

ans = 0

agc → 5   cg = 0

{ a c g a d g a g }

0	1	2	3	4	5	6	7
a	c	g	a	d	g	a	g
ans += cg		cg++	ans += cg		cg++	ans = ans + cg	cg++
ans = 3+3		cg = 3	ans = 3		cg = 2	ans = 1	cg = 1

ans = 6

```
cg = 0        ans = 0
for (i = N-1; i >= 0; i--) {
    if (s[i] == 'g') {
        cg++
    }
    if (s[i] == 'a') {
        ans += cg
    }
}
return ans
```

TC: O(N)

SC: O(1)

count a = 0   ans = 0

0	1	2	3	4	5	6	7
{ a	c	g	a	d	g	a	g }
ca++		ans += ca	ca++		ans += ca	ca++	ans += ca
ca = 1		ans = 1	ca = 2		ans = 3	ca = 3	ca = 6

Microsoft

Q2) Leaders in an Array

Given an array, you need to find leaders in A[]

An element is a leader if it is strictly greater than all elements on its right.

A[N-1] is always a leader

	0	1	2	3	4	5	6	7
A[8] =	{ 3	2	4	5	2	1	3	0 }
	✗	✗	✗	✓	✗	✗	✓	✓

Ans : 3

	✗	✓	✓	✗	✗	✗	✓	✓
	7	6	5	5	5	5	2	
	↑	↑	↑	↑	↑	↑	↑	↑
	0	1	2	3	4	5	6	7
A[8] =	{ 5	7	6	1	-1	0	5	2 }
	✗	✓	✓	✗	✗	✗	✓	✓

Ans : 4

If an element is greater than max of all elements on its right then it is a leader

```
for (i=0; i<N; i++){
    # find the max on right
    # [i+1, N-1] → maxR
    if (A[i] > maxR){
        leader++
    }
}

return leader
```

TC: $O(N^2)$

SC: $O(1)$

leaders = 3 max = 0

0	1	2	3	4	5	6	7
3	2	4	5	2	1	3	0

```
if (5 > max)        if (3 > max){
    leader++            leader++
    max=5               max=3
```

```
leaders = 1       max = A[N-1]
for (i=N-2; i>=0; i--){
    if (A[i] > max){
        leader++
        max = A[i]
    }
}
return leaders
```

TC: $O(N)$

SC: $O(1)$

	0	1	2	3	4	5	6	7
A[i] = {	7	7	6	1	-1	0	5	9
$m = -\infty$	7	7	7	7	7	7	7	$\boxed{9}$

Max in array

$min = \infty$

Integer. min value

inf

INT_MIN

```
curmax = -∞
for (i=0; i<N; i++){
    if(A[i] > curmax){
        curmax = A[i]
    }
}
```

→

```
curmax = -∞
for (i=0; i<N; i++){
    curmax = max(curmax, A[i])
}
```

Break (10:32 – 10:42)

Subarrays continous part of array

→ Can single element be a subarray? Yes

→ Can entire array be a subarray? Yes

→ Can 0 elements be a subarray? No

{2 3 5 7 1 0 -3}

{3} ✓

{2 3 5 7 1 0 -3} ✓

{3 5 7 1} ✓

{2 5 7} ✗

Closest Min Max

Amazon

Given an array find the length of smallest subarray which contains both min and max of array

Min : 1 Max : 6

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

Ans

Ans : 4

Min : 1 Max = 6

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

Ans : 3

Obs1) In the min len subarray, min ele & max ele will be at corners

sub1 $\underbrace{[\text{min} \; .... \; \text{max}]}_{L}$ ① ✓

sub2 $\underbrace{[\text{min} \; ... \; \text{max}}_{L} \; ....]$ ② ✓

{1 2 3 5}

Min Max

{1 2 3 5 4 2}

Min Max

{1 2 3 5 4 2}

2)

[Min Max] [Max Min]

1) Find Min, Max

3 3 5 7 3 3 3 3 3 3 3 7

min = 1 , max = 6

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

4

len = 3

$[2, 5] = 5 - 2 + 1$

$= 4$

-5

Max ⟶ Min

minlen = 4

Min ⟶ Max   TODO:

1) Find and max   TC: $O(N^2)$

min len = +∞   SC: $O(1)$

iterate over all the elements

→ if it is not min or max

skip

→ if it is max

$l = i$

look for min index on right $[i+1, N-1] \rightarrow r$

len = $r - l + 1$

minlen = min(minlen, len)

→ if it is min

$l = i$

look for max index on right $[i+1, N-1] \rightarrow r$

len = $r - l + 1$

minlen = min(minlen, len)

1) Min, Max values $\rightarrow$ O(N)

2) mini = -1 maxi = -1

3) minlen = ∞ / N

Min $\longrightarrow$ max on right

Max $\longrightarrow$ min on right

max = 6 min = 1

0	1	2	3	4	5	6	7	8	9	10	11
1	6	4	5	1	5	2	6	4	2	1	5

mini = 0
len = 2
maxlen = 2

maxi = 1
len = 4
minlen = 4

mini = 4
len = 4
minlen = 4

maxi = 7
len = 10 - 7 + 1
= 4
minlen = 4

mini = 10

[2, 2, 2, 2]

min = 1 max = 6

maxi = -1 mini = -1 minlen = ∞/N

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

maxi = 2

len = 4

mini = 5

len = 4

maxi = 8

len = 10-8+1

= 3

mini = 10

minlen = 3

r - l + 1

↑

r > l

maxi = -1 , mini = -1 , minlen = ∞

1) min_v max_v

```
for (i = n-1 ; i >= 0 ; i--) {
    if (A[i] == max) {
        maxi = i
        if (mini != -1) {
            len = max(mini, maxi) - min(maxi, mini) + 1
            minlen = min(minlen, len)
        }
    }

    if (A[i] == min) {
        mini = i
        if (maxi != -1) {
            len = max(mini, maxi) - min(maxi, mini) + 1
            minlen = min(minlen, len)
        }
    }
}
```

$\boxed{r > l}$

return minlen

$$\boxed{1 \le N \le \underline{10^6}}$$

$$\boxed{10^4 \le N \le 10^6}$$

5 second.