Q→ Given a character array, calculate no. of pairs s.t. $(i<j)$ & $s[i]$ = 'a' & $s[j]$ = 'g'.

In input all characters are lowercase. (a, b, c, d ... z)

s = [b a a g d c a g]
0 1 2 3 4 5 6 7

(1, 3) (2, 3) (6, 7)
(1, 7) (2, 7)

Ans = 5

s = [g c a g g a a]
0 1 2 3 4 5 6

(2, 3) (2, 4) Ans = 2

Bruteforce → ∀ i, j pairs check if $(i<j)$ & $s[i]$ = 'a' & $s[j]$ = 'g'.

TC = $O(N^2)$ SC = $O(1)$

```
ans = 0
for i → 0 to (N-2)
    for j → (i+1) to (N-1)

        if ( s[i] == 'a' && s[j] == 'g' )
            ans++
return ans
```

i = N-1

j → (N-1) + 1 = N Error! ✗

Observations → 1) ∵ given $i<j$ ∴ start j from $(i+1)$. → TC = $O(N^2)$

```
for (j = i+1 ; j <= (N-1) ; j++) {
     ✗              ✓
  . . . .
}
```

2) ∵ $s[i]$ == 'a' ∴ only loop j when i^{th} char is 'a'.

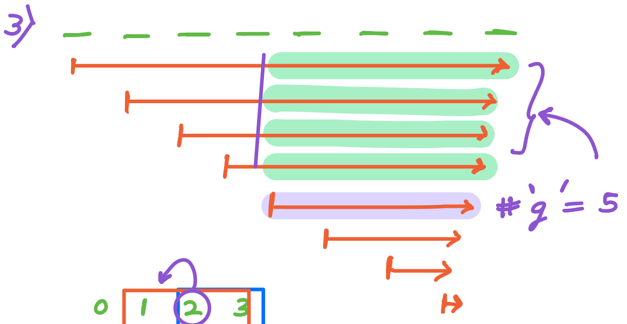
```
ans = 0
for i → 0 to (N-2)
  if(s[i] == 'a')
    for j → (i+1) to (N-1)
      if(s[j] == 'g')
        ans++
return ans
```

→ counting #'g' from index (i+1) to (N-1). ✓

TC = $O(N^2)$ SC = $O(1)$

s = [a a a a a --- a]

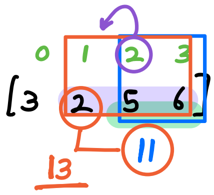
3)

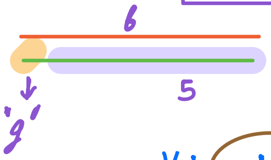


0	1	2	3	4	5	6	7
g	c	a	g	g	a	a	g
4	3←	3←	3	2	1←	1←	1

s = [g c a g g a a g]
cnt = [4 3←3←3 2 1←1←1]

```
cnt[i] = cnt[i+1]
if(s[i] == 'g')   cnt[i] += 1
```





Sum of elements from i to (N-1) ∀i.

```
∀i cnt[i] = 0
if (s[N-1] == 'g')
      cnt[N-1] = 1
for i → (N-2) to 0
  cnt[i] = cnt[i+1]
  if(s[i] == 'g')
      cnt[i] += 1
```

TC = $O(N)$
SC = $O(N)$

```
if(s[i] == 'g')
    cnt[i] = 1 + cnt[i+1]
else
    cnt[i] = cnt[i+1]
```

// cnt[i] → count of 'g' from i to (N-1).

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

s = [g c a g g a a g]
cnt = [4 3 3 3 2 1 1 1]

```
ans = 0
for i → 0 to (N-1)
  if(s[i] == 'a')
      ans += cnt[i]
return ans
```

TC = $O(N)$
SC = $O(1)$

Ans = 3+1+1 = 5 ✓

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

$TC = O(N+N) = O(N)$

$SC = O(N+1) = \boxed{O(N)}$

```
ans = 0
cnt = 0
for i → (N-1) to 0
    if (s[i] == 'g')
        cnt += 1
    else if (s[i] == 'a')
        ans += cnt
return ans
```

✓	✓	✓	✓	✓	✓	✓	✓
0	1	2	3	4	5	6	7

s = [g c a g g a a g] ←

cnt = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

ans = 0 + 1 + 1 + 3

= 5

$TC = O(N)$

$SC = O(1)$

Carry Forward

R to L

10:27 PM

Q→ Given an integer array A, count the number of leaders in the array.

Leader → An element which is greater than all elements on right of it.

→ Note → A[N-1] is always a leader.

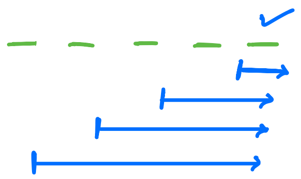
A =	[15	-1	7	2	5	4	-2	3]
	0	1	2	3	4	5	6	7
	✓	✗	✓	✗	✓	✓	✗	✓

Ans = 5

Leader = A[i] if (A[i] > max element from (i+1) to (N-1))

Bruteforce → ∀i check if A[i] > max element on right.

$TC = O(N^2)$ $SC = O(1)$



	0	1	2	3	4	5	6	7
A =	[15	-1	7	2	5	4	-2	3]
max →	15	7	7	5	5	4	3	3
	✓		✓		✓	✓		✓

ans = ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

```
ans = 1
m = A[N-1]
for i → (N-2) to 0
    if(A[i] > m)
        ans += 1
        m = A[i]
return ans
```

TC = O(N)
SC = O(1)

Subarray → continuous part of array.

$$A = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ [15 & -1 & 7 & 2 & 5 & 4 & -2 & 3] \end{array}$$

Single Element → ✓

Complete Array → ✓

Empty Array → empty subarray

Q → Given an integer array, find the length of smallest subarray which contains both min & max of array.

$$A = \begin{array}{ccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ [1 & 2 & 3 & 1 & 3 & 4 & 6 & 4 & 6 & 3 & 5] \end{array}$$

min = 1

max = 6

Ans = 4 (length)

A = [8 8 8 8 8]

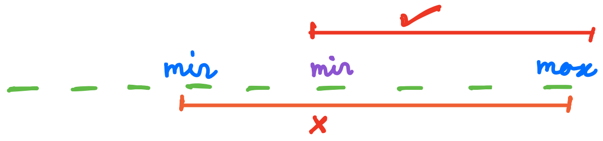
min = 8

max = 8

Ans = 1

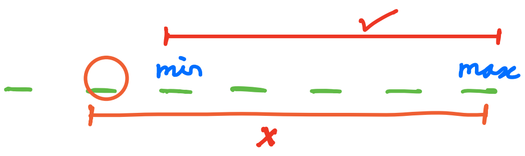
Observation → 1) Can smallest subarray containing min & max have more than 1 min element or max element?

No



2) Can the ans subarray have element in corner which is neither min nor max?

No



✓ [min ———— max] → find nearest min on left.

find nearest max element on right. ✓

✓ [max ———— min] → find nearest max on left.

find nearest min on right side. ✓

} H.W ✓

$$\begin{array}{lccccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ A = [& 2 & 2 & 6 & 4 & 5 & 1 & 5 & 2 & 3 & 6 & 4 & 3 & 1 & 2 & 6 & 3 & 6\,] \end{array}$$

min = 1 } TC = O(N)

max = 6 } SC = O(1)

(i, j)

(2, 5)

(5, 9)

(9, 12)

(12, 14)

Ans (length) → 5 − 2 + 1 = 4

9 − 5 + 1 = 5

12 − 9 + 1 = 4

14 − 12 + 1 = $\boxed{3}$ ✓

[L R] = R − L + 1

$$\begin{array}{lcccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ A = [& 2 & 6 & 1 & 1 & 3 & 6\,] \end{array}$$

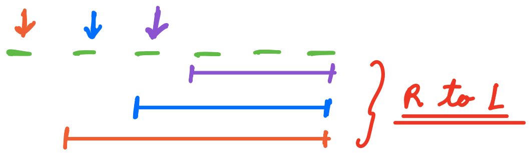
min = 1

max = 6

Ans = 2

Sol1 → ∀i if A[i] is min or max travel right from i to get closest max or min. ✓

TC = $O(N^2)$ SC = O(1)



$$\begin{array}{lccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ A = [& 2 & 6 & 4 & 1 & 5 & 2 & 6 & 3 & 4 & 6 & 5 & 3 & 2 & 1 & 1\,] \\ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ \end{array}$$ ←

min = 1 minId = ~~-1~~ ~~14~~ ~~13~~ 3

max = 6 maxId = ~~-1~~ ~~9~~ ~~6~~ 1

	length
(9, 13)	13 − 9 + 1 = 5
(6, 13)	13 − 6 + 1 = 8
(3, 6)	6 − 3 + 1 = 4
(1, 3)	3 − 1 + 1 = ③

Ans

TC = O(N) SC = O(1) ✓

ans = N → 1

→ minId = −1 maxId = −1

// calculate minA , maxA ✓ → 10

```
for i ⟶ (N-1) to 0
  → if(A[i] == minA) {
          minId = i     → 0
          if(maxId != -1)   x
               ans = min(ans, maxId - minId + 1)
    }  → if(A[i] == maxA) {
          maxId = i     → 0
          if(minId != -1) ✓
               ans = min(ans, minId - maxId + 1)
                        1 ,    0 - 0 + 1  = 1 ✓
    }
return ans
```

[8 8 8 8 8] ✓

[10]
0

i = 0