

Ques 1 Given a string  $s$ , count no. of pairs s.t

Amazon

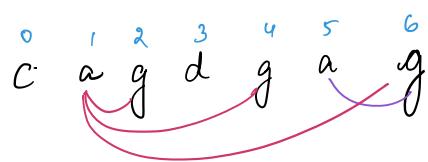
1)  $i < j$

2)  $s[i] = 'a'$

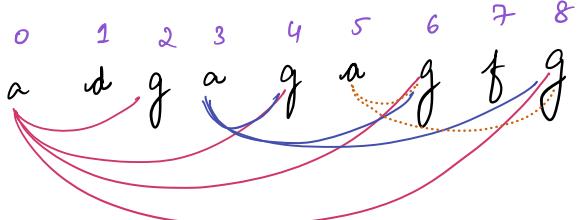
3)  $s[j] = 'g'$

Eg: 

$[0, 3]$   
 $[0, 5]$   
 $[4, 5]$   
 $[4, 3] \times$



$[1, 2]$   
 $[1, 4]$   
 $[1, 6]$   
 $[5, 6]$



$[0, 2]$   
 $[0, 4]$   
 $[0, 6]$   
 $[0, 8]$   
 $[3, 4]$   
 $[3, 6]$   
 $[3, 8]$   
 $[5, 6]$   
 $[5, 8]$

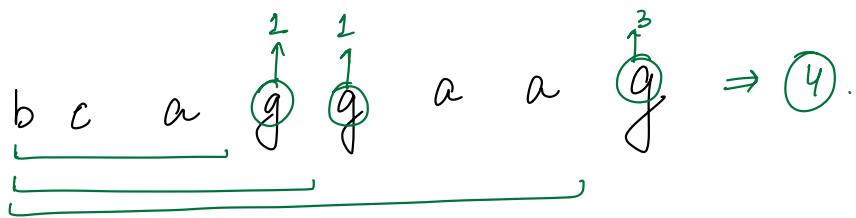
QUIZ 1 acgdgag

QUIZ 2 bcaaggaaag

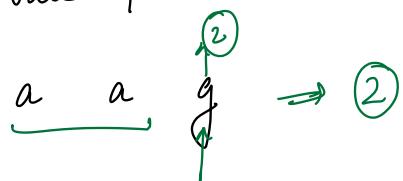
## Brute Force

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

TC  
 $O(n^2)$



Every 'g' will make a valid pair with all the 'a' on the left of it.



If we have count of 'a's till that particular 'g', we'll get answer for that 'g'.

variable count(a)	a	c	b	a	g	K	a	g	g
	1	1	1	2	2	2	3	3	3

ans → 0 2 5 8

ans = 0

count\_a = 0

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

Technique:  
Carry forward

TC  $\rightarrow O(|S|)$

SC  $\rightarrow O(1)$

We are finding no. of a's from left till a particular 'g'. In a similar manner, we can find no. of g's from right till a particular 'a'.

a	c	b	a	g	K	a	g	g
count_g	3	3	3	3	3	2	2	2 1

$$\text{ans} = \emptyset \neq 8$$

# write code for this  
on your own.

## Ques 2: Leaders in the array

Amazon  
Paypal

Given an array  $A$  of  $N$  integers. Find all leaders.

A leader is strictly greater than all ele to its right.

Ex ①  $A = [1, 2]$

Leader count  $\rightarrow 1$ .

#last ele will be counted  
as a leader always.

Ex ②  $A = [16, 17, 17, 4, 3, 5, 2]$

Leader count  $\rightarrow 1 \cancel{, 2} (3) \checkmark$

QUIZ 1    8, -2, 4, 7, 6, 5  
QUIZ 2    10, 7, 9, 3, 2, 4

Observation  
All ele to the right should be strictly smaller.

When I am at an ele, if I already know that all ele to its right are strictly smaller, that ele is a leader.

How to know whether all ele to right are smaller?

a    b    [c    d    e]  
↑              If I know the max  
                  element from c to e

If max-ele  $< b$   
then  $b$  is leader.

DRY RUN :  $A = [16, 17, 17, 4, 3, 5, 2]$

ans: 3

17

5

max = 2

Count = 1

2

3

ans

## Closest Min Max

Given an array A of N integers

Find smallest continuous part of array st it contains both the min & max value of the array

$$\text{arr}[] = [ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & \underline{1} & 3 & 4 & 6 & 4 & 6 & 3 \end{matrix} ] \quad \boxed{1 \ 3 \ 4 \ 6}$$

min: 1  
 max: 6

ans:  $6 - 3 + 1 = 4$

QVIZ 2 2 6 4 5 1 5 2 6 4 |

max → 6  
min → 1

ans ⇒ (8, 10) ⇒ 3

$$\text{arr}[] = [ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 2 & 8 & 4 & 5 & 2 & 5 & 2 & 8 & 4 & 2 \end{matrix} ] \quad \boxed{2 \ 8}$$

min: 2  
 max: 8

ans:  $8 - 1 + 1 = 2$

$$\text{arr}[] = [ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & " \\ 1 & 6 & 4 & 2 & 7 & 7 & 5 & 1 & 3 & 1 & 1 & 5 \end{matrix} ] \quad \boxed{7 \ 5 \ 1}$$

min: 1  
 max: 7

ans:  $7 - 5 + 1 = 3$

Observation \* check above array ele

① MIN & MAX are always the last elements of that sequence.

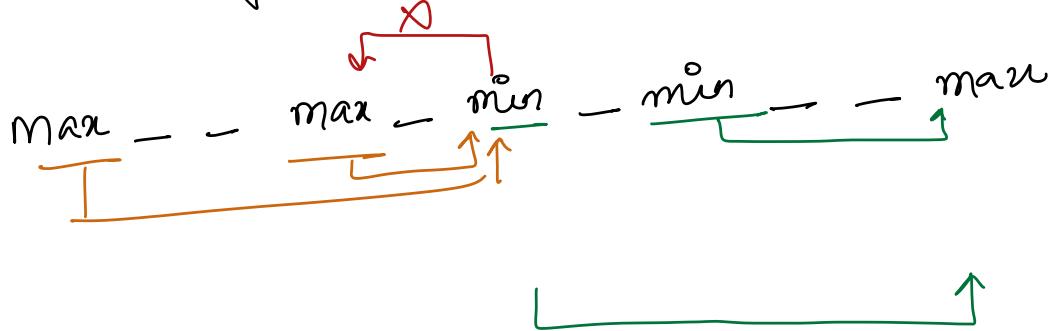
② Exactly one MIN & one MAX in that sequence

## Boute Force

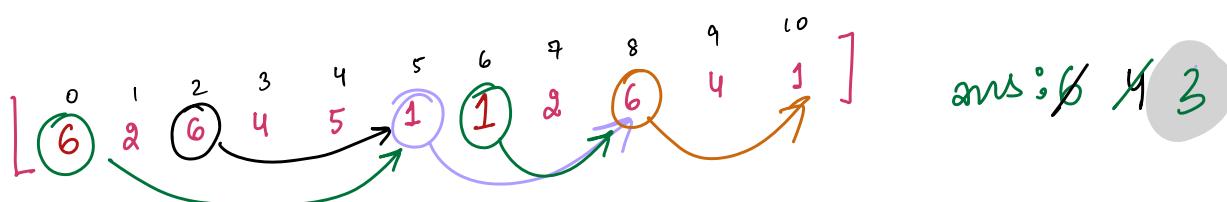
For every MIN, get the closest MAX on right

AND

For every MAX, get the closest MIN on right



Even if u move in one direction that's sufficient.



// for every min, find closest max on right  
 // for every max, find closest min on right  
 // find min & max of array →  $A_{\max}$  &  $A_{\min}$

```

for(i=0; i<n; i++) {
    if(A[i] == Amin) {
        for(j = i+1; j<n; j++) {
            if(A[j] == Amax) {
                ans = max(ans, j-i+1)
                break
            }
        }
    }
}
    
```

else if(A[i] == A<sub>max</sub>) {

```

        for(j = i+1; j<n; j++) {
            if(A[j] == Amin) {
                ans = max(ans, j-i+1)
                break
            }
        }
    }
}
    
```

TC

O(n<sup>2</sup>)

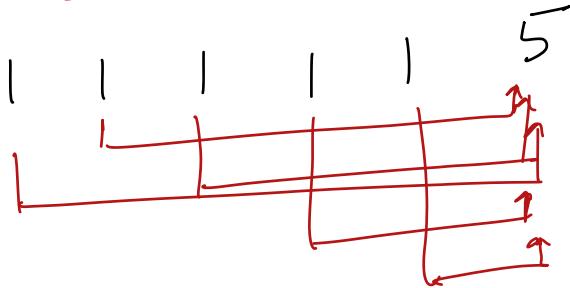
Q12      

0	1	2	3	4
8	8	8	8	8

$\min \rightarrow 8$   
 $\max \rightarrow 8$        $\underline{ans \rightarrow 1}$

Q12

TC



Observation :

①

is this check needed? → NO!

6 . . . 6 . . . 1

1 . . 1 . . 6

X

We just need to maintain the latest  
min encountered till now & latest max  
encountered till now.

max -- max - min - min -- max

| |

$\left[ \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 2 & 3 & 1 & 7 & 6 & 1 \end{array} \right]$ 
 last\_max = -1  
 last\_min = -1  
 length

$\left[ \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 7 & 2 & 1 & 1 & 2 & 7 & 5 & 3 & 1 & 7 & 6 \end{array} \right]$

$\left\{ \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 2 & 6 & 6 & 2 & 1 & 5 & 2 & 6 & 4 & 1 \end{array} \right\}$

$\left\{ \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 1 & 6 & 6 & 2 & 1 & 5 & 2 & 6 & 4 & 1 \end{array} \right\}$

# Write code by yourself  
 You can do it! 88

Q: Given  $N$  bulbs connected by a wire.  
Every bulb has two states  
 $\rightarrow 0$  (on)  
 $\rightarrow 1$  (off)

Directi | Ola | Amazon  
Cisco

Initial state is given.  
Wiring is faulty bcz of which if we press any switch, all bulbs with index  $> i$  change their state.

→ Find min no. of switches to press in order to switch on all bulbs.

Eg       $\begin{matrix} 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 0 & 1 \end{matrix}$

ans.       $\begin{matrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 \end{matrix}$       ④

QUIZ      1 0 1 0 1

Did anyone tried finding answer from right?  
Let's see what happens!

QUIZ      1 0 0 0 0 1

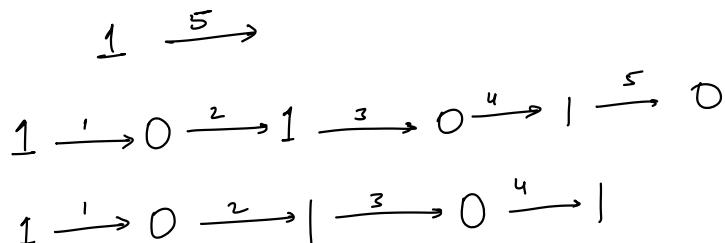
$T_C \rightarrow O(n^2)$

1	0	1	0	1	0	
1	1	0	1	0	1	
1	1	1	0	1	0	
				0		
				1		

on left some button  
got pressed

# Only two values are possible  
0 or 1.

Given no. of button presses before we reach a specific index, can we identify its current state?



0  $\xrightarrow{3\text{ or }1}$  1.      # no. of odd state changes  
# no. of even state doesn't change

	1	0	1	0	0	1	1	0
1st	1	1	0	1	1	0	0	1
2nd	1	1	1	0	0	1	1	0
3rd	1	1	1	1	1	0	0	1

Maintain  $\rightarrow$  no. of switch pressed.

```
switch-pressed = 0
for(i=0; i<N; i++) {
    //curr state
    if (switch-pressed % 2 == 0) {
        curr-state = A[i]
    } else {
        curr-state = 1 - A[i]
    }
    if (curr-state == '0') {
        switch-pressed ++
    }
}
return switch-pressed
```

0	1	2	3	4	5
1	0	0	1	0	0

switch-pressed = ~~0~~  
~~1~~  
~~2~~  
3