$ps[i] = ps[i-1] + A[i]$

$(0-i)$ $(0-i-1)$

$x \to y$ $\quad pf[y] - pf[x-1]$

Q array of N integers. For every index i,

$leftmax[i] = max[0-i]$

0	1	2	3	4	5	6	7	8	9
-3	6	2	4	5	2	8	-9	3	1
leftmax	-3	6	6	6	6	6	8	8	8	8

Basic:- for every index i, traverse from $0-i$ & find max elt

```
for (i=0; i<n; i++)
{
    for (j=0; j<=i; j++)
    {
    }
}
```

T.C: $O(N^2)$

$lm[0] = arr[0]$

$lm[1] = max(arr[0], arr[1])$

$lm[2] = \underbrace{max(arr[0], arr[1])}_{lm[1]}, arr[2])$

$lm[3] = \underbrace{max(a[0], a[1], a[2])}_{lm[2]}, a[3])$

$lm[3] = max(lm[2], arr[3])$

$$leftmax[i] = max(leftmax[i-1], arr[i])$$
$[0-i]$

↳ $leftmax[0] = arr[0]$

T.C: O(N)

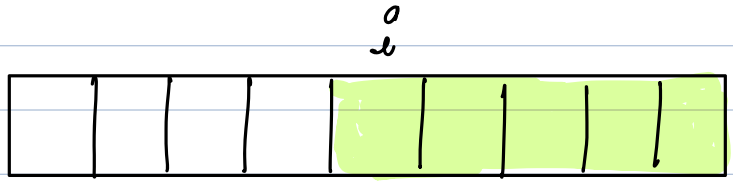
S.C: O(N)

↓

if you consider the output array as add" space

create this array

$rightmax[i] = max[i, n-1]$



~~$arr[0] = arr[0]$~~

$rm[N-1] = arr[N-1]$

$rm[N-2] = max(arr[N-1], arr[N-2])$

$rm[N-3] = max(arr[N-1], arr[N-2], arr[N-3])$

$rm[N-3] = max(rm[N-2], arr[N-3])$

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

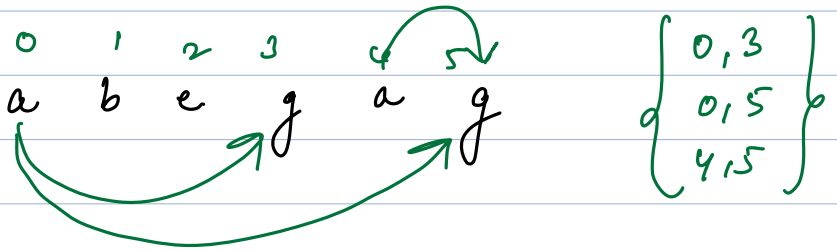
$rightmax[i] = max(rightmax[i+1], arr[i])$

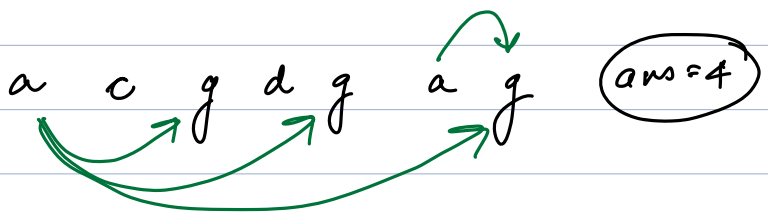
$rightmax[n-1] = arr[n-1]$

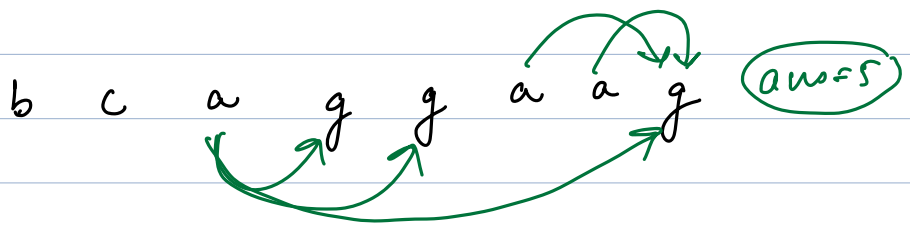
Q Given a string of lowercase english alphabets

Find the count of pairs (i, j) — indexes

$i < j$ && $arr[i] = 'a'$
$arr[j] = 'g'$







Brute: consider all pairs

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

T.C: $O(N^2)$
S.C: $O(1)$

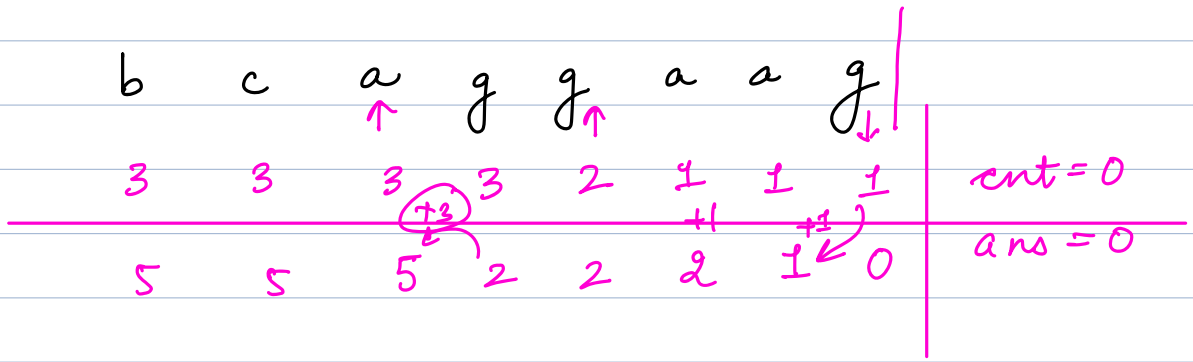
'a' - - - 'a'

count of 'g' on right side

OR

'g' - - - - 'g'

count of 'a' on my left



cnt = 0, ans = 0

```
for (int i = n-1; i >= 0; i--)
    if (arr[i] == 'g')
        cnt++;
    else if (arr[i] == 'a')
        ans += cnt;
}
return ans;
```

carry forward

T.C : O(N)

S.C : O(1)

10:25

Q ≡ given an array of size N. Find the length of shortest range which contains both min & max of complete array.

Ans ≡

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

min = 1
max = 6

ans = 6 - 3 + 1
= 4

circular is not allowed!

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

min = 1
max = 6

ans = 10 - 8 + 1 = 3

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

min = 1
max = 7

0	1	2	3	4	5
8	8	8	8	8	8

min, max

ans = 1

min __ max ____ min ___ min __ max __ max __.

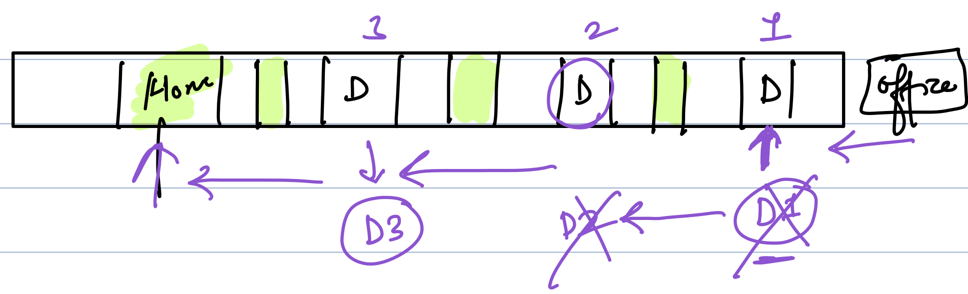
ans = ∞ / INT_MAX / N+1

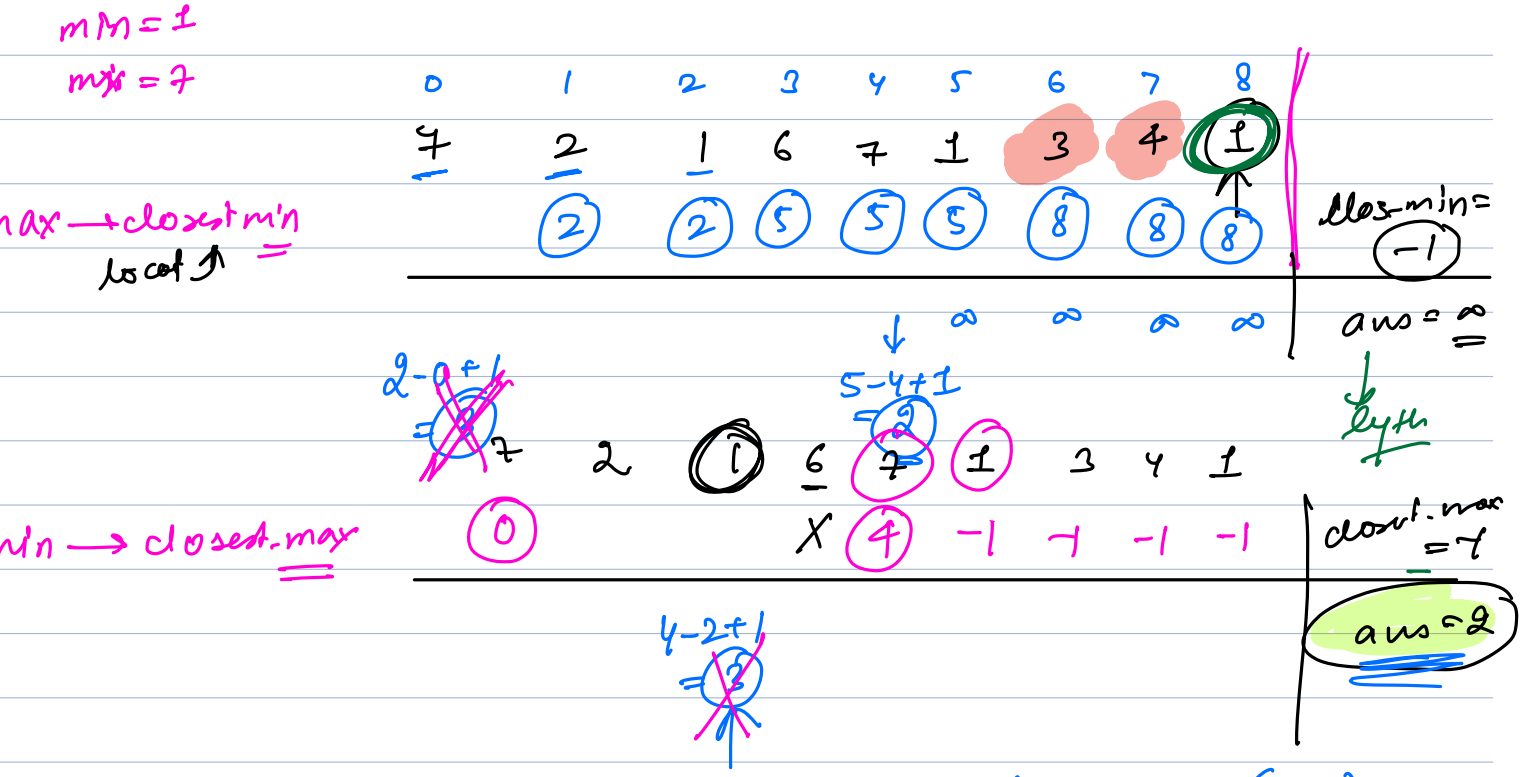
```
// find min & max !
for( int i=0; i<n; i++)
{
    if( arr[i] == min)
    {
        for(j=i; j<n; j++)
        {
            if(arr[j] == max)
            {
                ans = min(ans, j-i+1);
                break;
            }
        }
    }
    else if( arr[i] == max)
        find closest min
}
```

T.C: $O(N^2)$

S.C: $O(1)$

- min → closest max on right
- max → closest min on right






```
// min
// max
for( i = n-1; i >= 0; i--)
{
    if( arr[i] == min)
    {
        closest_min = i;
        if( closest_max != -1)
        {
            // update ans (j-i+1);
        }
    }
    else if( arr[i] == max)
    {
        closest_max = i;
        if( closest_min != -1)
        {
            // update ans;
        }
    }
}
```

```
if( min == max)
    return 1;
```