

18) Count Pairs "ag"

Given a $\text{char}[]$, calculate no of pairs i, j such that

$i < j$ & $s[i] = 'a'$ & $s[j] = 'g'$ All characters are lower case
 $[a, b, \dots, z]$

Exm: $0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$ } $\{ \begin{matrix} \text{Pairs : } 2, 3, 4, 6, 5 \\ \langle 1, 3 \rangle \ \langle 1, 7 \rangle \ \text{ans} = 5 \\ \langle 2, 3 \rangle \ \langle 2, 7 \rangle \\ \langle 3, 6 \rangle ? \ \langle 6, 7 \rangle \end{matrix}$

b	<u>a</u>	<u>a</u>	<u>g</u>	<u>d</u>	<u>c</u>	<u>a</u>	<u>g</u>
-----	----------	----------	----------	----------	----------	----------	----------

Q1: $0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$ } $\{ \begin{matrix} \langle 2, 3 \rangle \ \langle 3, 4 \rangle \\ \langle 5, 7 \rangle \ \langle 6, 7 \rangle \\ \langle 2, 7 \rangle \end{matrix} \} \text{ans} = 5$

<u>b</u>	<u>c</u>	<u>a</u>	<u>g</u>	<u>g</u>	<u>a</u>	<u>a</u>	<u>g</u>
----------	----------	----------	----------	----------	----------	----------	----------

Q2: $0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6$ } $\{ \begin{matrix} \langle 0, 2 \rangle \ \langle 0, 4 \rangle \ \langle 0, 6 \rangle \\ \langle 5, 6 \rangle \Rightarrow \text{ans} = 4 \end{matrix}$

<u>a</u>	<u>c</u>	<u>g</u>	<u>d</u>	<u>g</u>	<u>a</u>	<u>g</u>
----------	----------	----------	----------	----------	----------	----------

Idea: \Rightarrow Check all pairs (i, j)

$\text{cnt} = 0 \xrightarrow{\quad} \text{TC: } O(N^2) \text{ SC: } O(1)$

$i = 0; i < N; i++ \{ \xrightarrow{\quad} i = N-1; \{ j = N; j < N; j++ \}$

$j = i+1; j < N; j++ \{ \xrightarrow{\quad} \text{no error}$

$\{ \text{if } (s[i] == 'a' \text{ & } s[j] == 'g') \{ \text{cnt}++ \} \}$

// lca2: cut = 0

→ TC: $O(N^2)$

SC: $O(1)$

$i = 0; i < N; i++\{$

 if ($s[i] == 'a'$) {

$j = i + 1; j < N; j++\{$

 if ($s[j] == 'g'$) { cut++ }

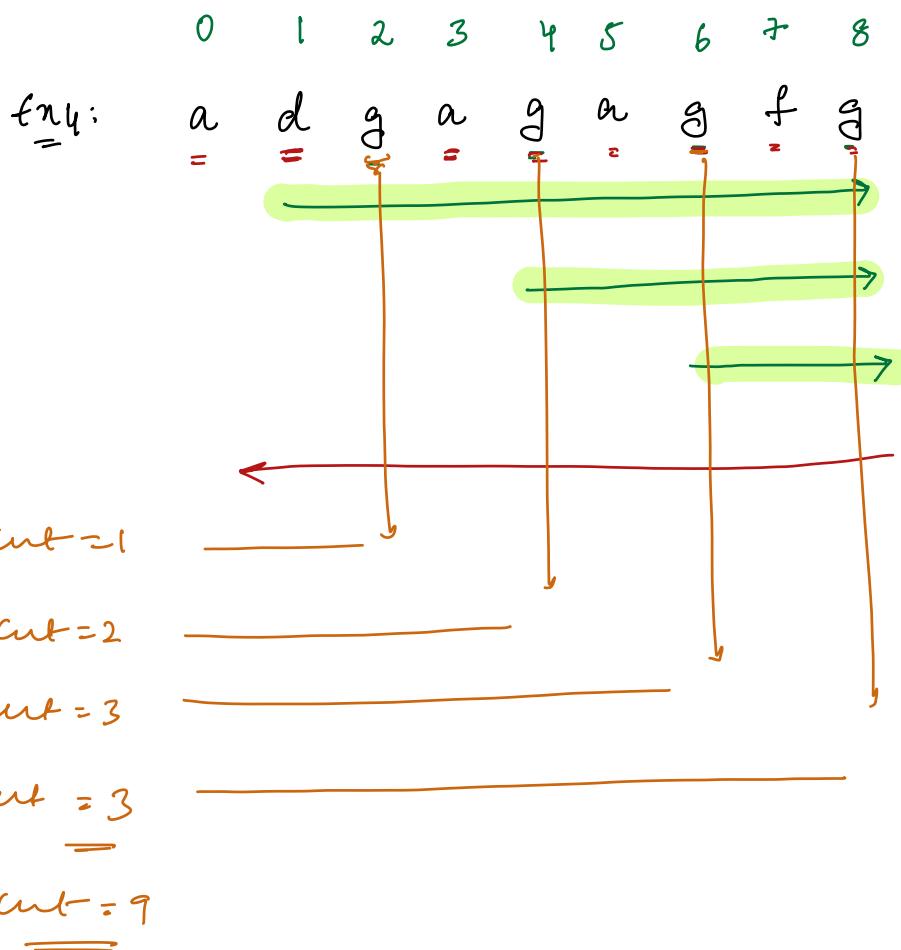
 if ($s[i] == 'a'$) {

 we are counting

no. of gc in

right side

cut = 0



cut = 4

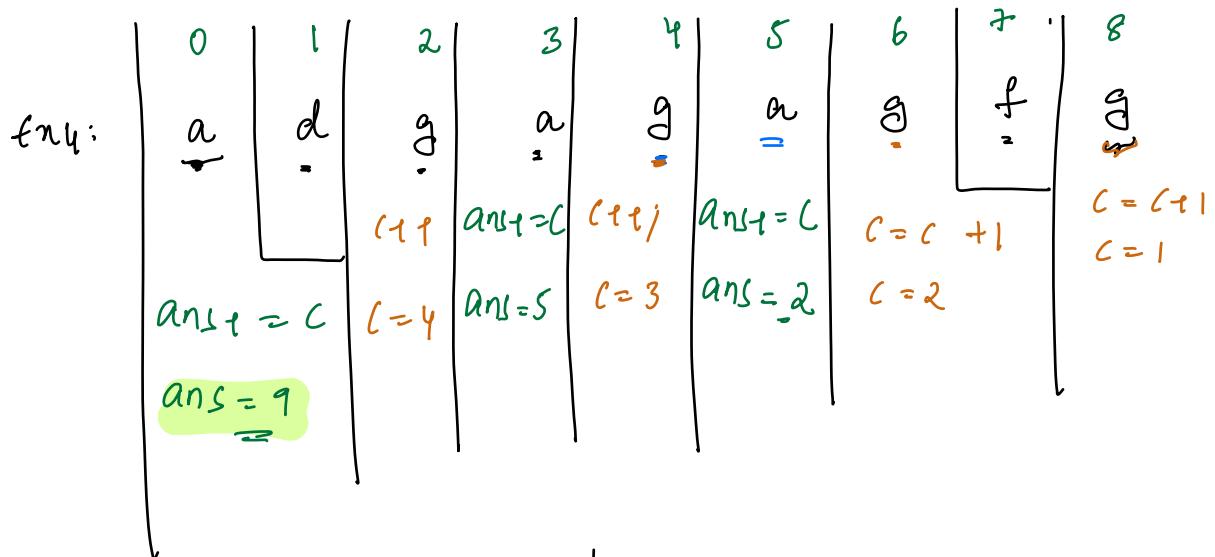
cut = 3

cut = 2

cut = 9

$\text{ans} = 0$, no. of pairs

$c = 0$ } number of g's in right side



$c = 0, \text{ans} = 0$

$i = N-1; i \geq 0; i-- \{$

 if ($s[i] == 'g'$) {
 c++
 }

use if ($s[i] == 'a'$) {

 ans = ans + c
}

}

return $\text{ans};$

Tc: $O(N)$ Sc: $O(1)$

TODO: → Homework Assignment

For a given calculate no. of
a's from Left → Right

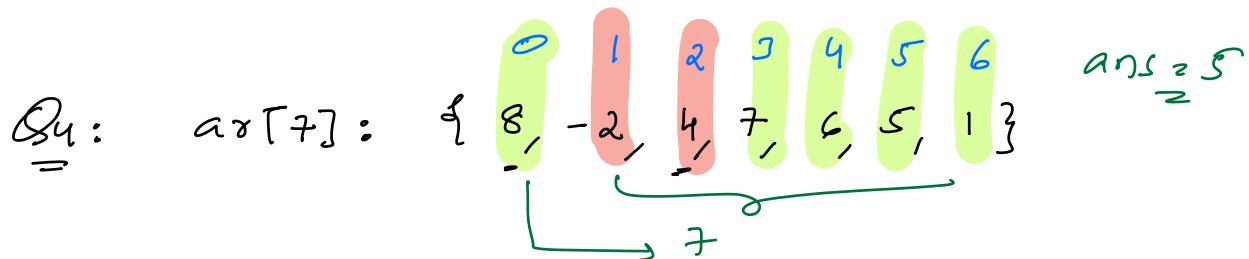
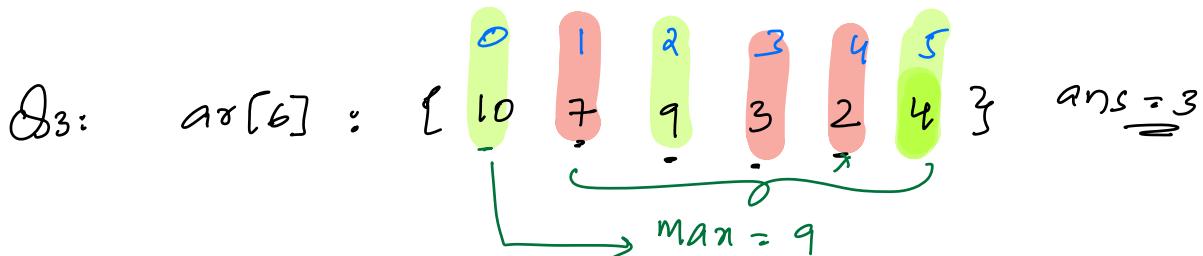
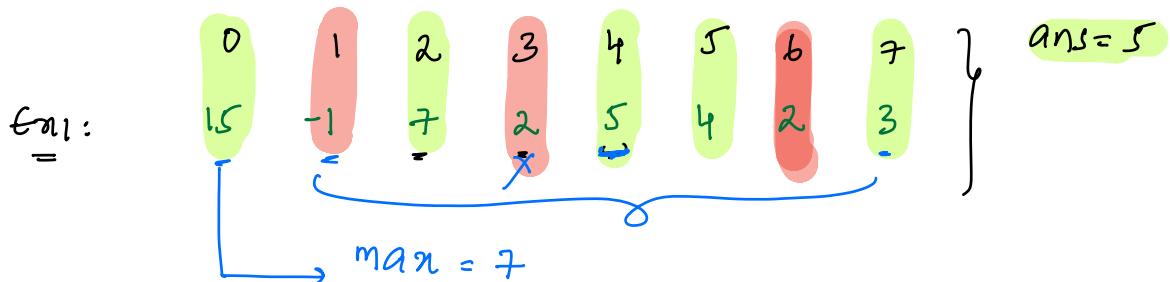
Tc: $O(N)$ Sc: $O(1)$

28) Leaders in a Array

Given an $ar[N]$, you have to find count leaders in arr]

An ele is leader, if it is strictly greater than max in entire right

Note: $ar[N-1]$ is always considered as leader



BF

D) For every element iterate
q right q get man.

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

```

i = 0; i < N; i++) {
    // iterate q get man
    // right
    j = i + 1; j < N; j++) {
        |
        |
    }
}

```

TC: $O(N^2)$

Idcc: Carry man R-L

cut = 1

man = $arr[N-1]$

$i = N-2; i \geq 0; i-- \{$

if ($arr[i] > man \{$

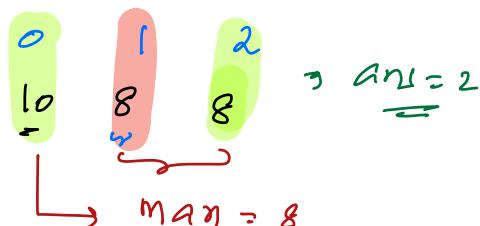
cut ++;

man = $arr[i]$

TC: $O(N)$

SC: $O(1)$

Ex1: $arr[] :$



Subarray Basics

1) Continuous part of an array is called Subarray

2) A single element is a Subarray

3) Full array is also Subarray

4) [-] → It is not Subarray

Ex: $\text{arr}[7] = -3 \ 4 \ 6 \ 2 \ 8 \ 7 \ 14 \ 9 \ 21$

Ex1: Indices $\rightarrow [2, \underbrace{3, 4, 5}]$: YES $\underline{[2, 5]}$ $\{2, 3, 4, 5\}$

Ex2: Indices $\rightarrow [3, 4, 6, 7, 8]$: $\swarrow 5$ is missing \rightarrow not Subarray.

Ex3: Indices $\rightarrow [1, 2, 3]$: YES

Ex4: Indices $\rightarrow [5]$: YES

Indices $\rightarrow [2-8]$ $\rightarrow [2, 3, 4, 5, 6, 7, 8]$: Subarray
 \swarrow length: $8-2+1=7$

Index $\rightarrow [s, e]$ \rightarrow length: $e-s+1$

Usage of predefined function

$\min(a, b) \rightarrow$ return min of 2 numbers
 $\max(a, b) \rightarrow$ return max of 2 numbers

$T_C: O(1)$

$T_C: O(N \log N) \leftrightarrow \text{Sort}()$ → sort the data

// Given N array get min?

$\min_val = arr[0]$

$i = 1; i < N; i++ \{$

$\} \min_val = \min(\min_val, arr[i])$

$\Rightarrow T_C: N \quad SC: O(1)$

$i: 40 \text{ Atm}$

} 2019 VNR CSE grad

→ Amazon SDE \Rightarrow 3 months

→ 2 years DSA

→ Scalar \Rightarrow 1 year 3 months

2020 September

3Q) Closest Min Max } } Given N Array Elements

Given an array find the length of Smallest Subarray which contains Both Min & Max of array?

Min: 1 Max: 6

Ex1: 0 1 2 3 4 5 6 7 8 9
1 2 3 1 3 4 6 4 6 3

Subarray: [3 - 8] : len = 6 } ans = 4

Subarray: [3 - 6] : len = 4

min = 1, max = 6

Ex2: 0 1 2 3 4 5 6 7 8 9 10
2 2 6 4 5 1 5 2 6 4 1

Subarray: [1 5] : len = 5 } ans = 3

Subarray: [2 5] : len = 4 } min = 8, max = 8

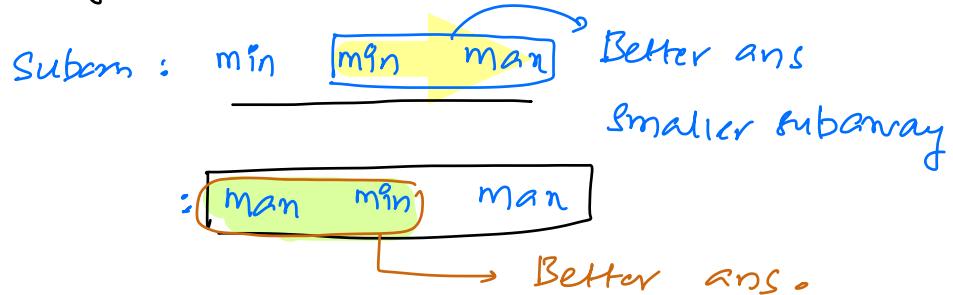
Subarray: [8 10] : len = 3 }

Ques3: 0 1 2 3 4
8 8 8 8 8

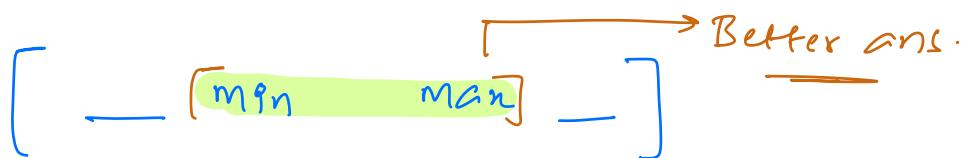
ans = 1

// observations, in your final ans subarray

1) We only need to have 1 min & 1 max



2) Min & Max should be present at corners?



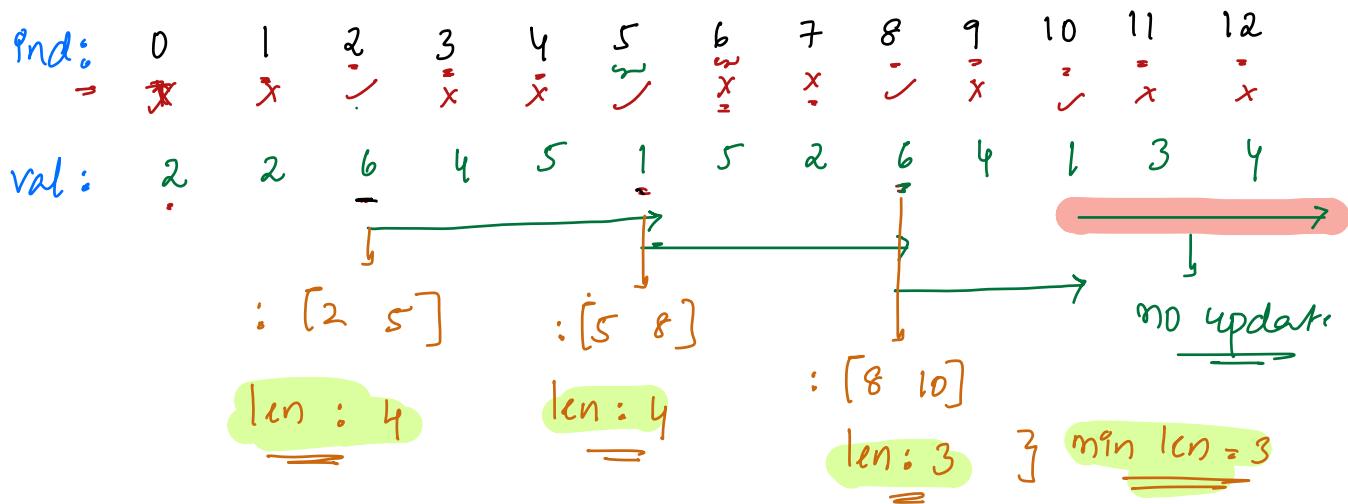
3) Case1: $\left[\boxed{\min \quad \max} \quad \right]$

Nearest max in right side

Case2: $\left[\boxed{\max \quad \min} \quad \right]$

Nearest min in right side.

Ex1: : min = 1, max = 6



Ex1: arr[5]: 10 3 2 6 1 } ans = 5

// Iterate q get min-val & max-val

if (min_val == max_val) return 1;

ans = N; \rightarrow TC: $O(N^2)$ SC: $O(1)$

optime =

i = 0; i < N; i++) { \rightarrow search for nearest max

if (ar[i] == min_val) { \rightarrow [i, j]

j = i+1; j < N; j++) {

if (ar[j] == max_val) {

ans = min(ans, j-i+1);

break;

}

\rightarrow search for nearest min.

if (ar[i] == max_val) { \rightarrow [i, j]

j = i+1; j < N; j++) {

if (ar[j] == min_val) {

ans = min(ans, j-i+1)

break;

}

return ans;

1) Iterate & get min-val & max-val
 2) if (min-val == max-val) return 1 -1 indicates min & max
 $\text{ans} = N, \text{mani} = -1, \text{min} = -1$ not found yet

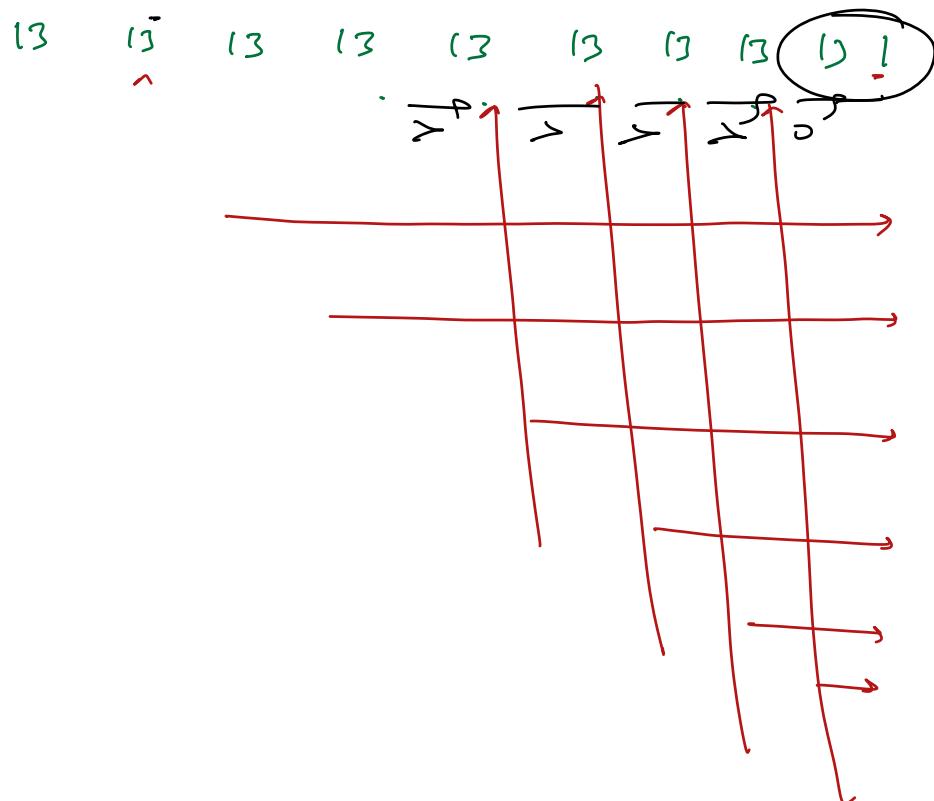
```

    p = N-1; q = 0; q-- {
      if (arr[i] == min-val) {
        mini = i;
        if (mani != -1) {
          ans = min(ans, mani - mini + 1)
        }
      }
      if (arr[i] == max-val) {
        mani = i;
        if (mini != -1) {
          ans = min(ans, mini - mani + 1)
        }
      }
    }
    return ans;
  
```

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

Doubt \Rightarrow According to above code:

Ex: lo: man = 13 min = 1



Iterate q \rightarrow get min-val & max-val

if (min-val == maxval) return 1;

ans = N; \Rightarrow TC $\in \underline{\mathcal{O}(N)}$

$i = 0; i < N; i++$ { \rightarrow search for nearer max

if ($ar[i] == \underline{\text{min_val}}$) { $\rightarrow [i, j]$ Extra Condition to make initial

$j = i+1; j < N; j++$ {

if ($ar[j] == \underline{\text{min_val}}$) { break; } $\rightarrow \mathcal{O}(N)$

if ($ar[j] == \underline{\text{max_val}}$) {

$ans = \min(\underline{ans}, j-i+1)$;

break;

} } \rightarrow search for nearer min.

if ($ar[i] == \underline{\text{max_val}}$) { $[i, j]$

$j = i+1; j < N; j++$ {

if ($ar[j] == \underline{\text{max_val}}$) { break; }

if ($ar[j] == \underline{\text{min_val}}$) {

$ans = \min(\underline{ans}, j-i+1)$

break;

return ans;

// Scenario

