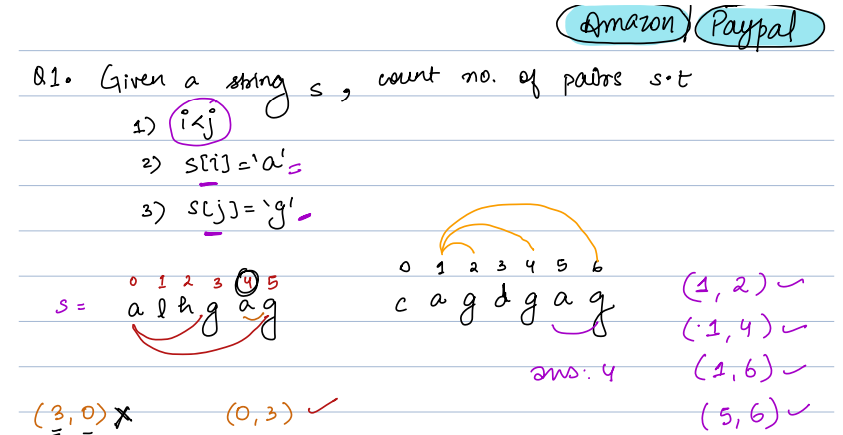
Q-1



Approach-1

Brute force:

Step1: Iterate A from 0 to n-1 in i

Step2: if A[i] != ‘a ’ then continue

Step3: Iterate j = i + 1 to n

Step4: if A[j] == ‘’g” then cnt = cnt+ 1;

Step5: return cnt;

TC = O(N\*N)

SC = O(1)

Approach-2 (Optimised):

Step1: iterate A from i to n

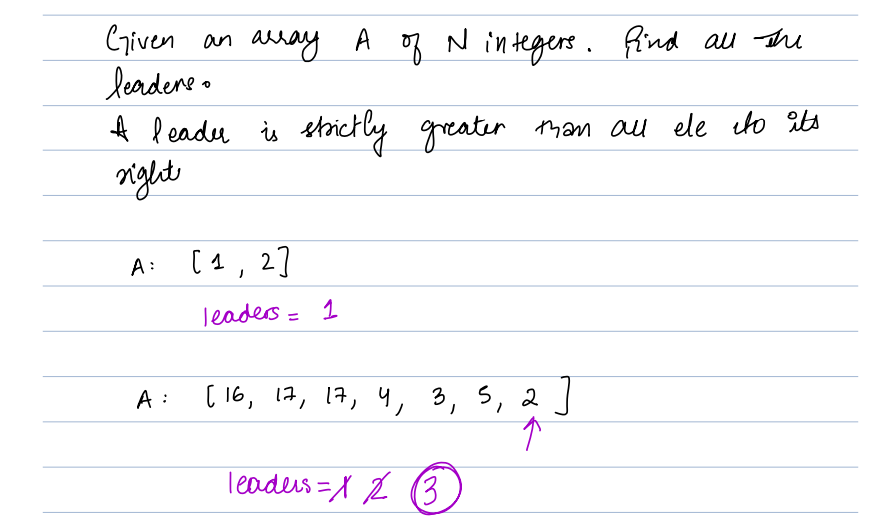
Step2: initial cnt = 0, acnt =0

Step3: if A[i] == ‘a’ then acnt ++

Step4: else if A[i] == ‘g’ then cnt = cnt + acnt;

Step5: return cnt;

Q-2:



Brute force:

TC = O(n\*n)

SC = O(1)

Optimised:

Step1: initialize currentLeader =A[n-1], List<int> leaders, leaders.Add(currentLeader);

Step2: Iterate A from n-1 to 0 in i

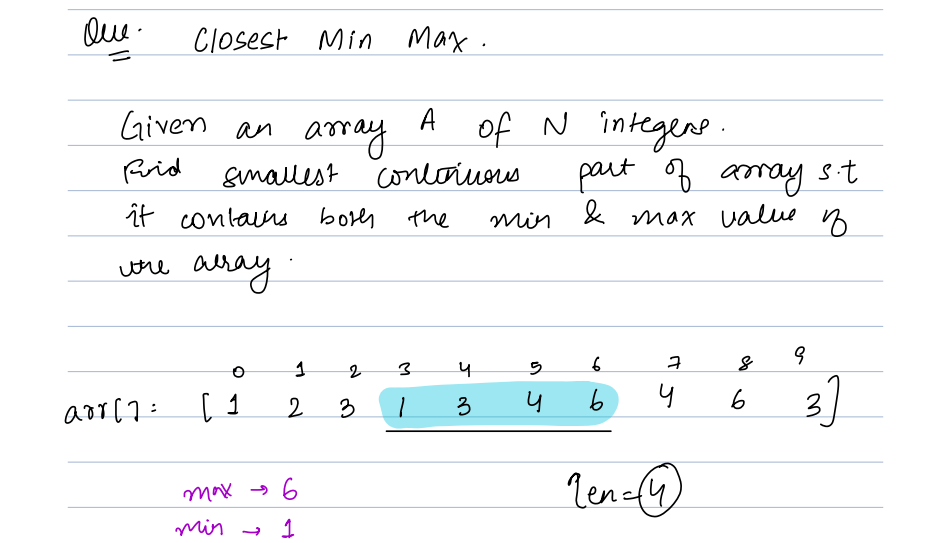
Step3: if currentLeader < A[i] then leaders.Add(A[i]);

Step4: return leaders;

TC = O(N)

SC = O(1)

Q-3:



Brute Force:

Optimised:

Step1: initialized minIndex , maxIndex = -1, ans = 0

Step2: find minValue and maxValue from A

Step3: iterate A from 0 to n in i

Step4: if A[i] == minValue then

{

minIndex = i

if (maxIndex != -1){

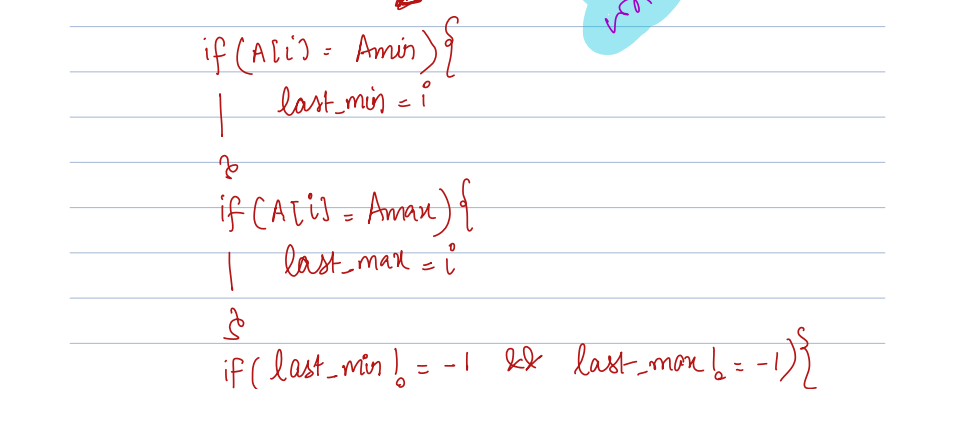
ans = min(ans, I – maxIndex + 1)

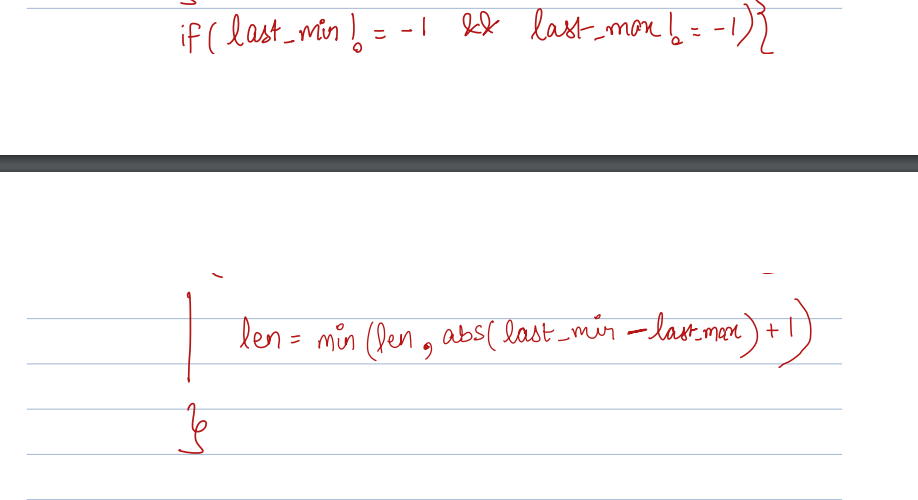
}

}

if A[i] == maxValue then maxIndex = I;

Step5: ans = min (ans, abs(maxIndex - minIndex ))





Step6: return ans;

Bulbs :

A wire connects**N**light bulbs.

Each bulb has a switch associated with it; however, due to faulty wiring, a switch also changes the state of all the bulbs to the right of the current bulb.

Given an initial state of all bulbs, find the **minimum number** of switches you have to press to turn on all the bulbs.

You can press the same switch multiple times.

**Note:** **0** represents the bulb is off and **1** represents the bulb is on.

i/p: A = [0, 1, 0, 1]

i/p 2: A = [0, 1, 0, 1, 0]

1-> [1, 0, 1, 0, 1]

2-> [1, 1, 0, 1, 0]

3-> [1, 1, 1, 0, 1]

4-> [1, 1, 1, 1, 0]

5-> [1, 1, 1, 1, 1]

o/p: 4

Cases1: Switch 1, change to [1, 0, 1, 0]

2, change to [1, 1, 0, 1]

3, change to [1, 1, 1, 0]

4, change to [1, 1, 1, 1]

Brute force:

For (i =0 ; I < n ; i++)

{

If (A[i] == 0) {

// toggle A[i]

A[i] = 1;

Count ++;

For(j = i+1; j < n; j++){

// toggle all element

A[i] = ~A[i]

}

}

}

Return count;

TC = O(n \* n)

SC = O(1)

Optimisation:

i/p = [0, 1, 0, 1]

[1, 0, 1, 0]

Cnt = 0;

(I = 0, i< n-1; i++)

{

If (cnt % 2 == 0) {

If (A[i] == 0){

Cnt++;

}

}else{

If (~A[i] == 0){

Cnt++;

}

}

}

TC = O(N)

SC = O(1)