Special String

Need help? Get on call with TA/Helper

Flag Question

Problem Description

A string having lowercase latin characters is considered special if it begins with some consecutive 'a's (atleast one) and then immediately contains some consecutive 'b's (atleast one) till the end (A string of the form **"a...b..."** is special).

In other words, a string is special if it is a concatenation of two non-empty strings S_1 and S_2 such that S_1 contains all 'a's and S_2 contains all 'b's.

Given a string **A**, you have to tell it is special or not.

NOTE: A special string must contain atleast one 'a' and atleast one 'b'.

Problem Constraints

$1 <= |A| <= 10^5$

Ex aaxbb. ans = No

Ex aaabbab ans = No

Ex aaabbb ans = Yes

Ex aaabbbaaabbb ans = No

Ex ab ans = Yes.

Approach: Check if any char other than 'a' & 'b' return false if so. else true

✗ not correct aaba.

Solution. 1. Count no of 'a' in front. c_1

2. Count no of 'b' in end. c_2.

$c_1 + c_2 = n$ (len of string)

```
fn ( String a )
{
    int c1 = 0, c2 = 0;

    for( int i = 0 ; i < n ; i++ )
    {
        if( a[i] == 'a' ) c1++;
        else break;
    }

    for( i = n-1 ; i >= 0 ; i-- )
    {
        if( a[i] == 'b' ) c2++;
        else break;
    }

    if( c1 > 0 && c2 > 0 && c1 + c2 == n ) return "Yes"
    else return "No";
}
```

max # of iteration = $n+n$ ✗

O(N)

$c_1 = 3$

$c_2 = 3$

"bb"

$c_1 = 0$ $c_2 = 2$

$c_1 + c_2 = 2$ n

aaabbba

$c_2 = 0$.

aaaxbbb.

$C_1 = 3$ $C_2 = 3$ $C_1 + C_2 = 6$

Q.

Sisters Love To Sing

Need help? Get on call with TA/Helper

Flag Question

Problem Description

Ananya and Bhavya are two loving sisters. They have many hobbies in common. Singing is one such hobby. They are wonderful singers.

The sisters love to sing. But there is one sad problem. When they sing at the same time, they do not coordinate very well, i.e. Ananya sings in high tones while Kavya sings in low tones and when they sing at same time, the resultant voice becomes sometimes awful to hear.

Now to solve this issue, they decided to sing at same tone when their performances overlap to give a heart-touching performance.

Ananya decided that she will start her performance at $\mathbf{A_1}$ minutes and end at $\mathbf{A_2}$ minutes from now, while Bhavya decided to start her performance at $\mathbf{B_1}$ minutes and end at $\mathbf{B_2}$ minutes.

They want to know how much time (in minutes) both of them will be singing simultaneously. i.e. number of minutes when both of them will be singing at the same time.

Problem Constraints

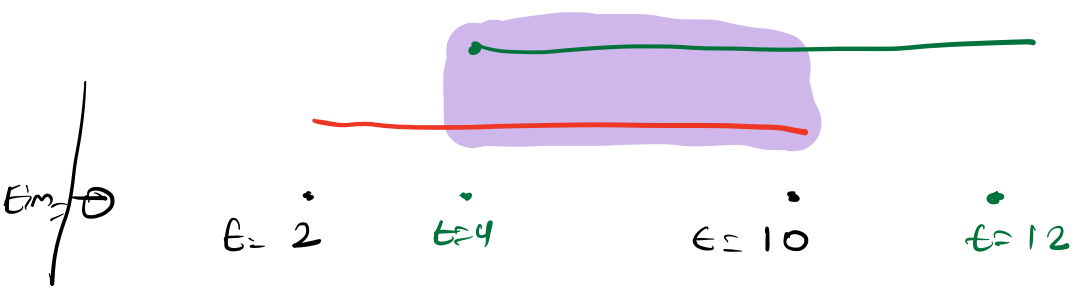
$0 <= \mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}, \mathbf{B_2} <= 10^9$

$\mathbf{A_1} < \mathbf{A_2}$

$\mathbf{B_1} < \mathbf{B_2}$

$[A_1, A_2]$ $[B_1, B_2]$

$[2, 10]$ $[4, 12]$



$10 - 4 = 6$ seconds.

ans = 6

t2 $A_1, A_2 = [3, 100]$

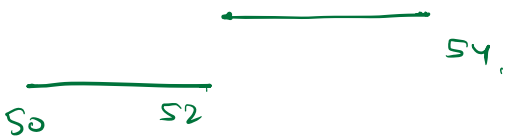
$B_1, B_2 = [1, 5]$

5 - 3 = 2 seconds, ans = 2,

Ex: $A_1, A_2 = [50, 52]$

$B_1, B_2 = [52, 54]$

ans = 0



Make Cases.

If $(A_1 \leq B_1)$ // First sister will start first.

- If $(A_2 \leq B_1)$ ans = 0
- else if $(A_2 \leq B_2)$ ans = $A_2 - B_1$
- else ans = $B_2 - B_1$

else

// Second sister is starting first.

swap(A_1, B_1) swap(A_2, B_2)

// Now, first sister is starting first

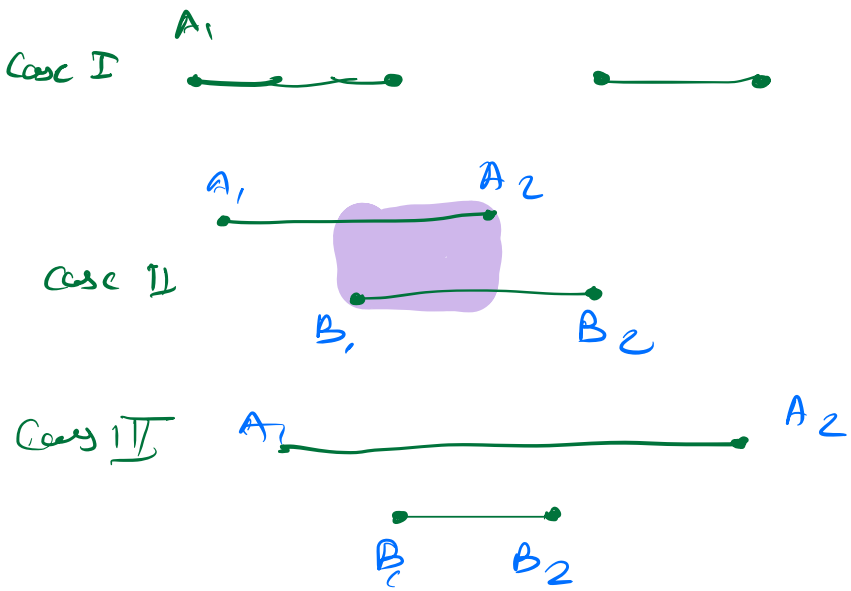
If ($A_2 \leq B_1$) ans = 0

else if ($A_2 \leq B_2$) ans = $A_2 - B_1$

else ans = $B_2 - B_1$

# of iteration = $1n^0$

O(1)



MINIMUM PICKS

Need help? Get on call with TA/Helper

Flag Question

Problem Description

You are given an array of integers **A** of size **N**.
Return the difference between the maximum among all even numbers of **A** and the minimum among all odd numbers in **A**.

Problem Constraints

2 <= N <= 1e5
-1e9 <= A[i] <= 1e9
There is atleast 1 odd and 1 even number in A

$1e^9 = 1 \times 10^9 = 10^9$

$2e^9 = 2 \times 10^9$

$$ans = Even_{max} - Odd_{min}$$

Ex 2, 3, 5, 1, 100, 99, 50.

100 − 1 = 99

Ex 10 12 14 15 17 19.

14 − 15 = −1.

−10000001 (Integer.INT_Value.)

Integer.INT_Max

$-1e^9 - 1$

int max_even = ___ mini_odd = 1e9+1

```
for( i = 0, i < N, i++ )
{
  if( a[i] % 2 == 0 )   max_even = max(max_even, a[i])
  else                  mini_odd = min( mini_odd, a[i] )
}
```

# of iteration = N,
O(N)

return max_even − mini_odd.

Even Remainder

Need help? Get on call with TA/Helper

Flag Question

Problem Description

Write a program to input an integer A and another array B of N numbers from user and print the count of elements of B such that B[i]%A is even.

Problem Constraints

1 <= A <= 100

1 <= N <= 100

1 <= B[i] <= 1000

int A

int[] B = size N.

Ex B = [2, 10, 11, 13, 14, 100]

A = 10

2 0 1 3 4 0

ans = 4.

int ans = 0

# of iterations = N

O(N)

for (i = 0, i < N ; i++)

{

int x = B[i] % A.

if (x % 2 == 0) ans ++

return ans.

Little Ponny and Maximum Element

Need help? Get on call with TA/Helper

Flag Question

Problem Description

Little Ponny is given an array, **A**, of **N** integers. In a particular operation, he can set any element of the array equal to **-1**.

He wants your help in finding out the minimum number of operations required such that the maximum element of the resulting array is **B**. If it is not possible, then return **-1**.

Problem Constraints

$1 <= |A| <= 10^5$

$1 <= A[i] <= 10^9$

Ex A = [1, 2, 8, 3, 5, 10]
B = 3.
-1 -1
ans = 2 ⇒ [1, 2, 3, 3, -1, -1] after 2 op.

Ex A = [1, 2, 3, 99, 45, 48]
B = 99.
ans = 0.

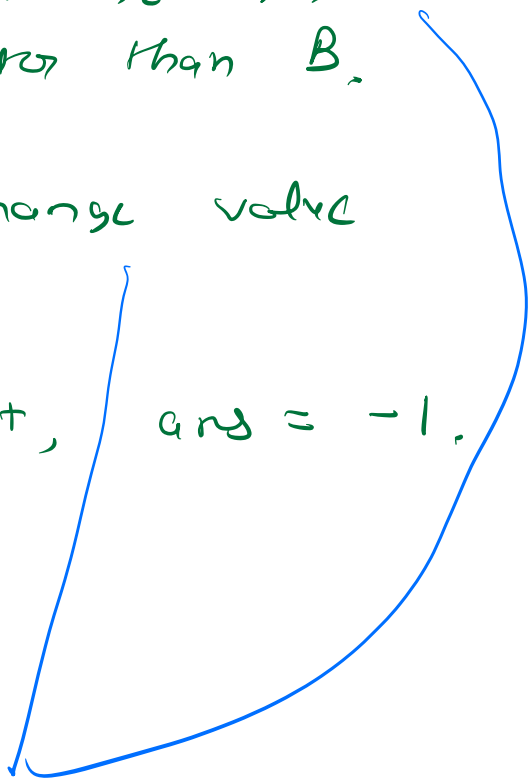
Ex A = [1, 4, 5].
B = 3.
ans = -1

Observation.

1. I need to change all the no. which are greater than B.

2. I only need to change value greater than B.

3. If B is not present, ans = -1.



ans = no. of greater than B elements.

```
int ans = 0, flag = 0
for( i = 0 to N-1)
{
    if( A[i] > B]  ans++
    if( A[i] == B]  flag = 1
}
if( flag == 0)  return -1   // B is not present.
return ans.
```

# of iterations = N.

O(N)

Doubt -

max_even ⟶ Store the max_even encounter so far.

for ( ) iteration 5th

```
for(
{
    if(a[i]%2==0)
    {
        max_even = max(max_even, a[i]);
    }
}
```

max even	a[i]	updated max_even
4	6	6
4	2	4