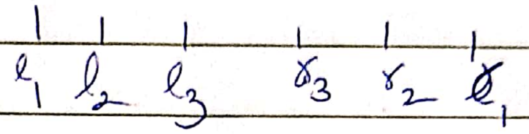
CF Div-2 602 __/__/____

(A) Including the leftmost (min) right & rightmost l (max) will give us the solution

In a solution like



As only one point will suffice so required length is 0,

In other cases where the answer is >0, we need to include 1 point from the right most range which will be it's l & one point from leftmost range that will be r. thus answer will be

$$\max(0, \overset{\max}{l} - \underset{\min}{r})$$

(B) If current element is more than the previous max we change the previous max & include this in the answer as change of maximum will be because of this element only.

If the max remains same then it means a number less than the previous max has come, So we check for the numbers smaller than the max which have not been included yet & include the one which has not been visited.

Now when is it impossible to create a permutation? When i > a[i]. ~~as there~~ For Ex,

Pos	1	2
	1	1

Now there is no number smaller than 1 to be put at Pos(1) except 1 but if we put it there, there is no way to put any other number at 2 such that 1 is still maximum.

|D| We can sort in descending order first (& for lexicographicaly smaller take the lesser of the indexes in case value of elements is same)

Then we can take n list of 1 to n sizes & sort them by index & return lis[K-1][pos-1].first to get respective values.

__/__/____

D2 We sort in descending order same as D1. None this list ~~does~~. X.

But as size is 2.10^5, we can't maintain different lists.

So we sort the queries on basis of K & then maintain a SegOst(implementation in good problems) where if query K size increases we insert as many elements as difference or change in size from X & for each query we simply findorder(pos) to get the index of element & take the value from original array.

Scanned by CamScanner