Playing with segments editorial

A set of k segments is good if there is a segment [Li, Ri] (1<=i<=k) from the set such that it intersects every segment from the set(insertion must be point or segment).

Let's iterate over this segment (which intersects all the others) and construct a good set of the remaining segments, maximum in terms of inclusion. It is easy to understand that this set will include all segments that intersect with ours. We must delete all other segments.

Two segments [l1,r1] and [l2,r2] intersect if max(l1,l2)≤min(r1,r2). Then if the segment that we iterate over has coordinates [L,R], then we must remove all such segments [l,r] for which r<L or R<l is satisfied (that is, the segment ends earlier than ours begins, or vice versa).

Note that these two conditions cannot be fulfilled simultaneously, since l<=r and if both conditions are satisfied then r<L<=R<l. This means that we can count the number of segments suitable for these conditions independently.

Each of these conditions is easy to handle. Let's create two arrays— all the left boundaries of the segments and all the right boundaries of the segments. Let's sort both arrays. Now we can count the required quantities using the binary search or prefix sums (but in this case, we need to use the coordinate compression technique).

Taking at least the number of deleted segments among all the options, we will get the answer to the problem.