



The Big Prize

Subtask 1

In this subtask, we can find the diamond with a simple binary search.

Subtask 2, Solution 1

First, we query the first 474 prizes in the line. These queries are sufficient to find at least one box containing lollipop. When we find it, we can binary search to find all prizes that are more expensive than lollipops. This approach needs less than 9000 queries. More specifically, it requires $O(\sqrt{n} \times \log n)$ queries.

Subtask 2, Solution 2

If we query boxes i and j ($i < j$) and both of them contain lollipops, we can find the number of boxes having more expensive prizes (than lollipops) in the boxes between i and j : Suppose Rambod tells there are x boxes to the left of box i that contain more expensive prizes. He also tells there are y boxes to the left of box j that contain more expensive prizes. Then, there are exactly $y - x$ boxes between the boxes i and j that contain more expensive prizes.

We consider the whole sequence of boxes as one segment. With a query of box i , we break the segment into two segments: the boxes before, and the ones after i (the boxes that are already queried are not in any segments). While doing binary search, we can track the number of boxes in each segment. If among our previous queries there are two boxes i and j with the above conditions to the right and to the left of a segment, we can ignore that segment and do not query that.

Subtask 2, Solution 3

While the previous solution can get full score of this task, we can further optimize it. The idea is that it is sufficient for the boxes i and j to have the same prize values to identify the boxes containing more expensive prizes between them. Further, we can sum the pair of

values answered by Rambod for each box. Two boxes i and j contain equal prizes if the sum of answered values for box i is equal to the sum of answered values for box j . Hence there is no need to look specifically for lollipops: we do a normal binary search, and we do not continue binary searching in a segment if we find such boxes i and j that show there is no more expensive prize in that segment. A further improvement is to see if $y - x$ is equal to the number of identified prizes between the boxes i and j that are more expensive to the prizes in the boxes i and j , we can ignore any segment between this pair of boxes. This method, if implemented efficiently, needs at most 4759 queries.