Analysis: Board Meeting

Test set 1

In the first test set, there is only one king. There are various ways for us to deduce its position; we will describe one that requires three queries and is relatively easy to understand.

First, we query (-1000000, 1000000), which is the upper left corner of the board; let A_1 be the result. It tells us which J-shaped region of the board the king is in, illustrated as follows starting from the upper left corner:

```
0123...
1123
2223
3333
 .      .
 .      .
 .      .
```

Next, if we did not happen to get lucky and find the king, we query (-1000000 + A_1, 1000000 - A_1). That is, we query the corner of that J-shaped region. Call the result A_2; it tells us how far away from the corner the king is, within that region. However, we do not know whether the king is to the left of or above the corner.

Finally, if we have still not found the king, we query (-1000000 + A_1 - A_2, 1000000 - A_1). That is, we guess that the king is to the left of the corner. If the result is 0, then we have found the king; otherwise, we know the king is above the corner, at (-1000000 + A_1, 1000000 - A_1 + A_2).

Test set 2

With more than one king, we find that it is not possible to determine the exact location of each king. For example, we cannot distinguish between a case with kings at (+1, 0) and (-1, 0) and a case with kings at (0, +1) and (0, -1). So it must be possible to answer queries with less complete information about the kings' locations.

The "L^∞" metric in the problem, for which we take the maximum of the absolute differences of two coordinates, is a little inconvenient to work with directly, which results in the fairly ad-hoc nature of the test set 1 solution above.

Converting to diagonal coordinates makes things simpler. If a point is at (x, y) in the original coordinates, let (u, v) = (x + y, x - y) / 2 be its diagonal coordinates. Then if the diagonal coordinates of two points are (u1, v1) and (u2, v2), the distance between them is |u1 - v1| + |u2 - v2|. So if the kings are located at diagonal coordinates (u_i, v_i) for i= 1, ..., N, then the result of a query for the point (u, v) is Σ_i (|u - ui| + |v - vi|) = (Σ_i |u - ui|) + (Σ_i |v - vi|)

We can handle these two sums separately. To compute the first one, we need to know which "downright-sloping" diagonal lines the kings lie on, and to compute the second one, we need to know which "upleft-sloping" diagonal lines the kings lie on.

Now we need a method to compute these diagonals. Consider the point (-2M, 0). This point is far enough to the left that, if we propose this point, the total distance we get will be the sum of the differences in x coordinates. If we propose (-2M, 1), we will get the same answer, unless there is a king at (-M, -M), in which case we will get an answer one higher.

More generally, for a positive integer K, let d(K) for a positive integer K be the difference in the results of proposals of (-2M, K) and (-2M, K-1). d(K) will be equal to the number of kings below the diagonal $x + y = -2M + K$. We can use binary searches to find the places where d(k) changes value, and the magnitudes of those changes, to find out which diagonals of the form $x + y = c$ contain kings, and how many they contain.

Similarly, by proposing points (-2M, -K), we can find out how many kings lie on the diagonals that go in the other direction. Once we have this data, we are able to compute the sums above and respond to queries from the judge.