A highway and the seven dwarfs

Author(s) of the problem: **Michal Forišek**
Contest-related materials by: **Ján Oravec, Richard Kráľovič**

First of all note that a highway is good iff it doesn't intersect the convex hull of all dwarfs' houses (if it does intersect the convex hull, it intersects its edge and the vertices of this edge are two houses lying on different sides of the highway). After reading the coordinates of the N houses we may pre-compute their convex hull, e.g. using some well-known algorithm running in $O(N \log N)$ time.

Now we will be given many lines and for each of them we have to decide, whether it intersects a given convex polygon. How to do this in a reasonable time? Suppose we have a line with some fixed direction. According to this direction, some of the vertices of the polygon is the leftmost one and some of them is the rightmost one. Clearly the line intersects the polygon iff these two points lie in different half-planes. Also it would suffice if we could find the leftmost and the rightmost point for a given direction.

There are a few different methods how to find them in $O(\log N)$ time, we will present one that involves some more pre-computation, but is in our opinion easiest to implement.

Suppose we have a fixed direction and we know the leftmost and the rightmost point for this direction. If we now rotate the direction by a small angle, it is very likely that the leftmost and the rightmost point won't change. Let's take a closer look on when does a change occur. We'll draw two lines with our direction, one of them passing through the leftmost point and one through the rightmost one.

When we rotate the direction, the leftmost and rightmost point won't change until one of the sides of the convex hull will lie on one of the two lines. In this moment the other end of this side will become the new left/rightmost point. Also the two points we seek change only when the direction of the line is the direction of one of the convex polygon's sides.

There polygon has $O(N)$ sides and their directions divide the set of all directions (e.g. angles of the line with the x-axis) into $O(N)$ intervals. For each of them we compute the leftmost and the rightmost point using the algorithm described above. This phase can be implemented in $O(N)$ time.

Now if we get a line, we simply compute its direction and use binary search to find out its interval (and the corresponding two points) in $O(\log N)$. If the number of lines is M, the total time complexity of this algorithm is $O\left((M + N) \log N\right)$.