Playing Go

Problem author: Alexey Mikhnenko, developer: Igor Markelov

First, let us look at the optimal answer. Obviously, the area of the answer is at least the area of the convex hull of the original set of points. In the case when the area is larger, it is claimed that the white point we moved is shifted along the normal to one of the $n \cdot (n-1)/2$ segments connecting the original points.

This observation allows us to write an $O(n^4 \cdot \log(n))$ solution — we iterate over the direction and the point, shift it, and recompute the optimal answer as the area of the convex hull of the resulting set.

Note that (of course, the proof of this amazing observation is left as a most interesting exercise for the curious reader) for each oriented direction it is sufficient to consider at most 5 points (the farthest in this direction). Thus, the previous solution turns into $O(n^3 \cdot \log(n))$ if for each direction we first choose at most 5 candidates and then build the convex hull for each of them.

Now let us learn how, for a fixed set of k points, to answer queries of the form «the area of the convex hull of the set if one more point is added» in $O(\log(k))$ time with $O(k \cdot \log(k))$ preprocessing. Build the convex hull on the k points, and compute prefix sums of the areas along the convex hull from the first vertex to the i-th one for all vertices. To answer a query, first use the standard algorithm in $O(\log(k))$ to check whether the point lies inside the convex hull. If it does — the answer is the area of the original hull. If it does not — find the tangents from the point to the convex hull; the answer area is equal to the area of the triangle formed by the tangent points and the query point, plus the area of the original convex hull without the part of the polygon cut off by the diagonal between the tangent points. The tangent search can be implemented in $O(\log(k))$, and the required areas can be computed in $O(1)$ using the precomputed prefix sums.

Apply this primitive to the previous solution: precompute the convex hulls of the original set with each of the white points removed, then the solution will work in $O(n^3)$, since the heaviest part is considering $O(n)$ candidate points for $O(n^2)$ directions.

A careful reader of the statement, having reached this point in their reasoning, may have appreciated the generous gesture from the jury — to remove from consideration test cases with three or more points on one line.

Let us optimize this part using the standard rotating sweep-line technique over directions. For $O(n^2)$ directions, we need to maintain a list of $O(n)$ points sorted along the direction. The total number of events will be $O(n^2)$, so now the solution will run in $O(n^2 \cdot \log(n))$.