I have three different algorithms—quick hull, brutal force and improved brutal force. I wrote the improved brutal force algorithm with brutal force concept, but realized that it is much better O(n^3). I run the test several times and average the runtime to produce the table in the excel sheet. All graphs in the document can be obtained by run the Plot class.

# Quick Hull

Quick hull method divide points into half set and then process them separately. By ignoring points inside the convex we have now, this algorithm reduce the size of points drastically.

Without considering those points ignored, its recurrence relationship should be

T(n) = 2\*T(n/2)+O(n)

which makes T(n) ⋲ O(nlogn), but in fact quick hull method is much faster due to those points eliminated. Similarly to quick sort, T(n) can be O(n^2) for some special input.

The experiment suggests that T(n) is more close to Θ(n^1/2) for the given data set. One pretty close match of the experiment result is



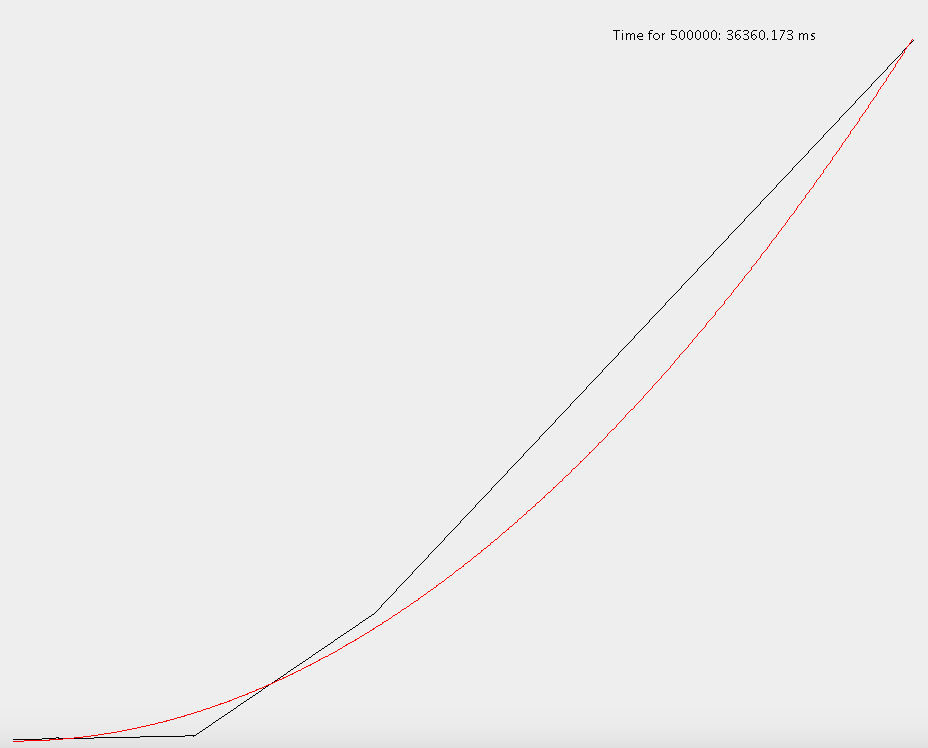
Shape of T(n) (black) and approximation (red) is on the right side.

Quick hull algorithm can process 90,000,000,000 points in one minute.

# Improved Brutal Force

Improved brutal force method first uses a brutal force method to find a seed in the set, which will be a vertex in the convex hull. This would take O(n^3) because it may need to search (n-m) points until finally find a valid point on the convex hull, and each points search would take at most (n-1)(n-2) comparisons. However, in real process, it only takes just a few points to determine that a pair of points is not valid for convex, so the actual runtime for this step is closer to Θ(n^2). Then start with this seed, search for the next vertex in the clockwise direction. The algorithm halts and returns the result when the convex is completed. For each vertex (m) needed to be searched, it needs to consider total (n-1) pairs involving the seed and check whether they produce a valid edge by checking whether all others (n) points are on the right side of this line. Therefore, the algorithm to grow the seed is O(mn^2), where m is the number of vertex in the final convex hull and n is the number of points. Hence, the over all algorithm is O(n^3) and Θ(n^2) theoretically.

The experiment indicates that improved brutal force is Θ(n^2), which is reasonable since m is very small compared to n. The closest approximation of experimentation is

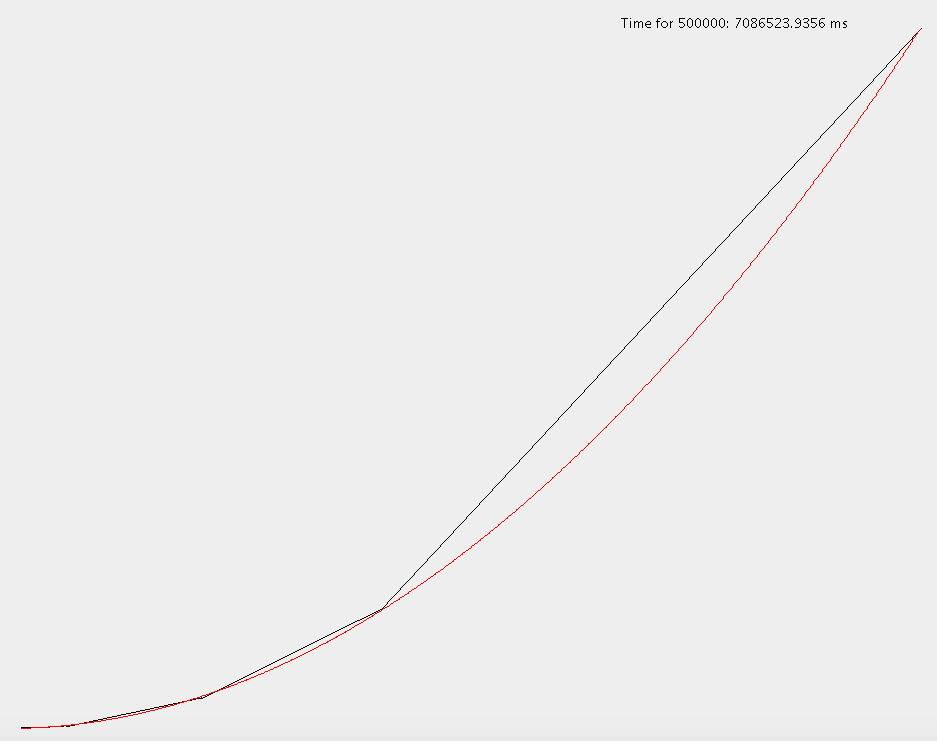


Shape of T(n) (black) and approximation (red) is on the right side.

Quick hull algorithm can process 641,199 points in one minute.

## Brutal Force

Brutal force method simply check all possible pairs (n(n-1)/2), and check whether other points (n) are on one side of it. As stated in the book, brutal force algorithm takes O(n^3). However, in actual implementation, to determine whether a point pair is not at the convex edge, we only need to find two points on different sides of line, so for most pairs checking total n points is unnecessary. Therefore, the actual average runtime is more close to Θ(n^2). Then with the possible points, run a search to take points only at vertices instead of on the edge. This step takes Θ(m).

The closest approximation of experimentation is



Shape of T(n) (black) and approximation (red) is on the right side.

Brutal force algorithm can process 40,204 points in one minute.

I did not receive code from anyone else or share your code for this project with anyone else.

Fred Zhang