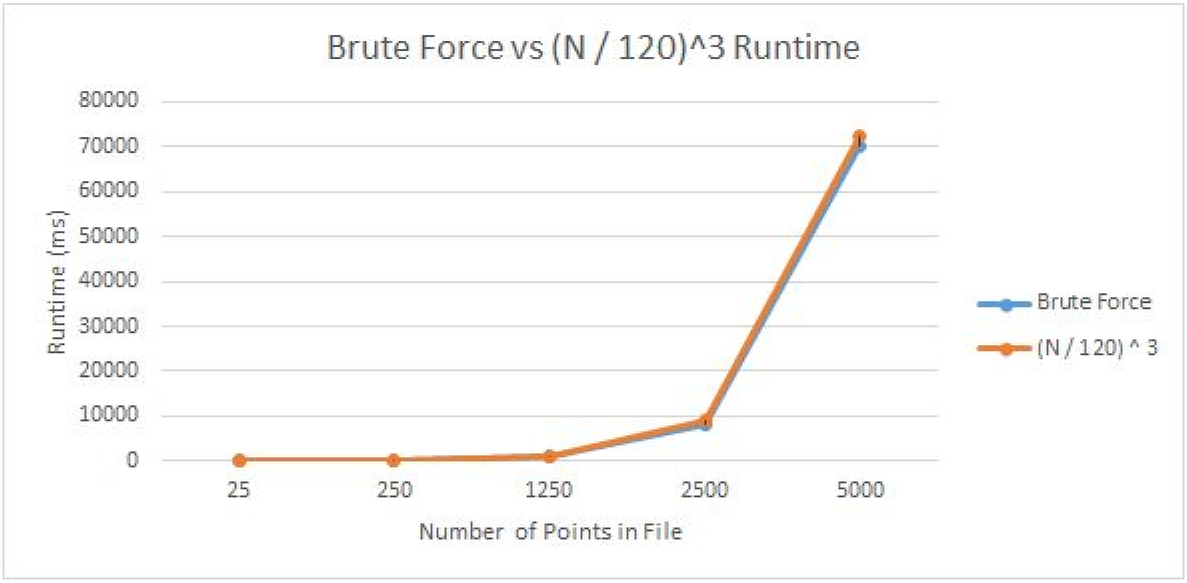
Austin Derrow-Pinion
MA/CSSE 473
Convex Hull Algorithm Time Analysis

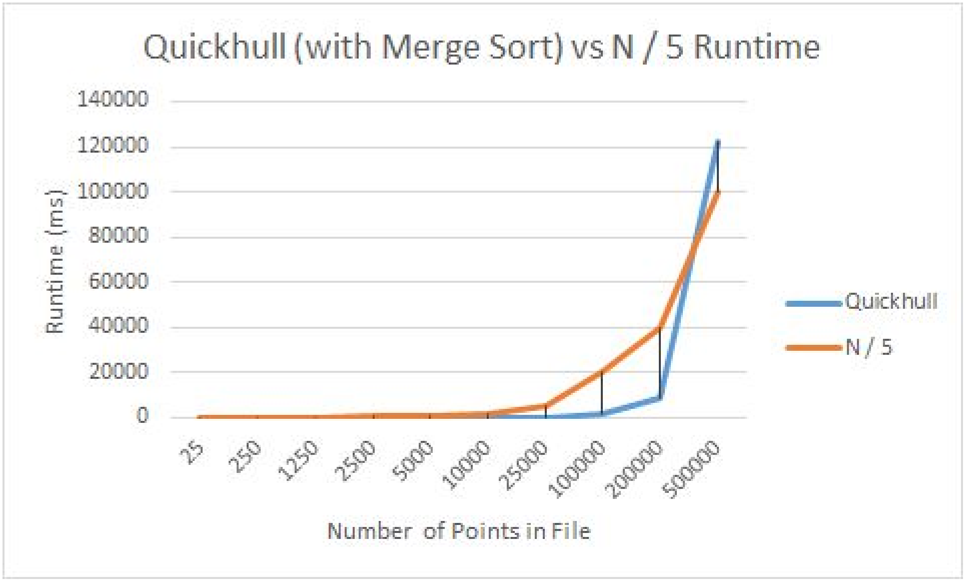
For the brute force method, I was able to calculate up to the file with 5000 points. Any more than 5000 points takes too long to calculate. This brute force algorithm has a runtime efficiency on the same growth rate as $O(N^3)$, therefore as expected, it grows very fast. Here is a graph of my results, the function fits the runtime almost perfectly.

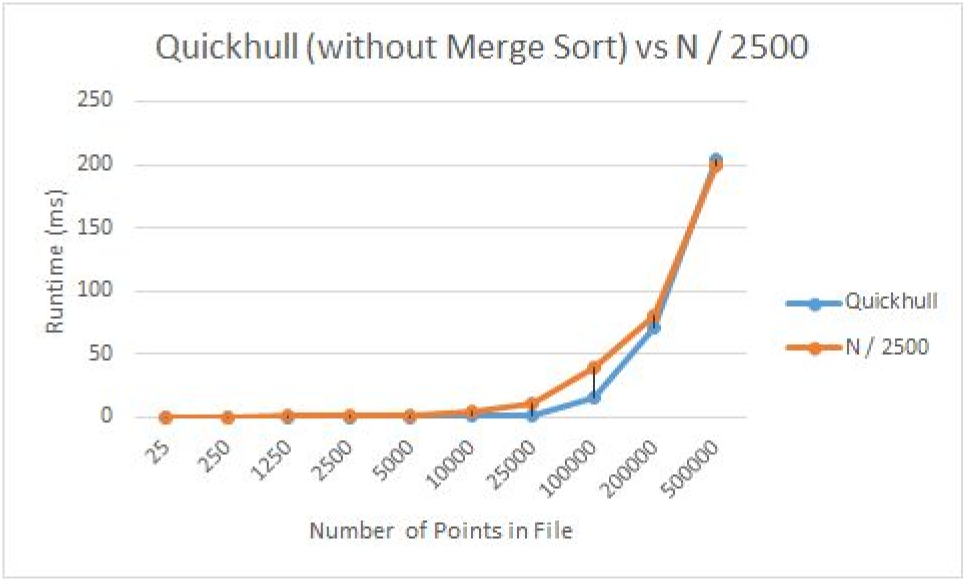


For the quickhull algorithm, I was able to calculate the runtime for every file. This algorithm was much quicker at solving the problem. By the proof located here,
http://surface.syr.edu/cgi/viewcontent.cgi?article=1058&context=eecs_techreports
the algorithm runs in $O(N^2)$ for worst case, but an expected runtime of O(N log(N)) when all vertices are extreme points. These input files (more so as the file size increased) were the exact opposite, where there were very little extreme points in comparison to the overall number of points. For a situation like this, the paper proves expected runtime will be O(N).

It was difficult to fit a line to the results when I included the merge sort algorithm in the computation of the runtime. Although you stated on Piazza to include this, I also tried it without the sorting in the calculation. I did for two reasons. First, in the book it states the algorithm assumes the points are sorted when given as input. Second, if the expected runtime is linear, the merge sort algorithm took up most of the time for large values of N since it has a O(N log(N)) runtime. I included a table with both values and two different graphs showing the difference. The graph for the time computed without the merge sort algorithm looks the best and fits to a linear function the best in my opinion.

| Number of Points | Quickhull (ms) | Quickhull without Merge Sort (ms) |
|---|---|---|
| 25 | 0 | 0 |
| 250 | 0 | 0 |
| 1250 | 2 | 0 |
| 2500 | 4 | 0 |
| 5000 | 8 | 0 |
| 10000 | 16 | 1 |
| 25000 | 45 | 2 |
| 100000 | 1659 | 16 |
| 200000 | 8484 | 71 |
| 500000 | 122599 | 204 |


Quickhull (with Merge Sort) vs N / 5 Runtime
Runtime (ms)
Number of Points in File
Quickhull
N / 5


Quickhull (without Merge Sort) vs N / 2500
Runtime (ms)
Number of Points in File
Quickhull
N / 2500