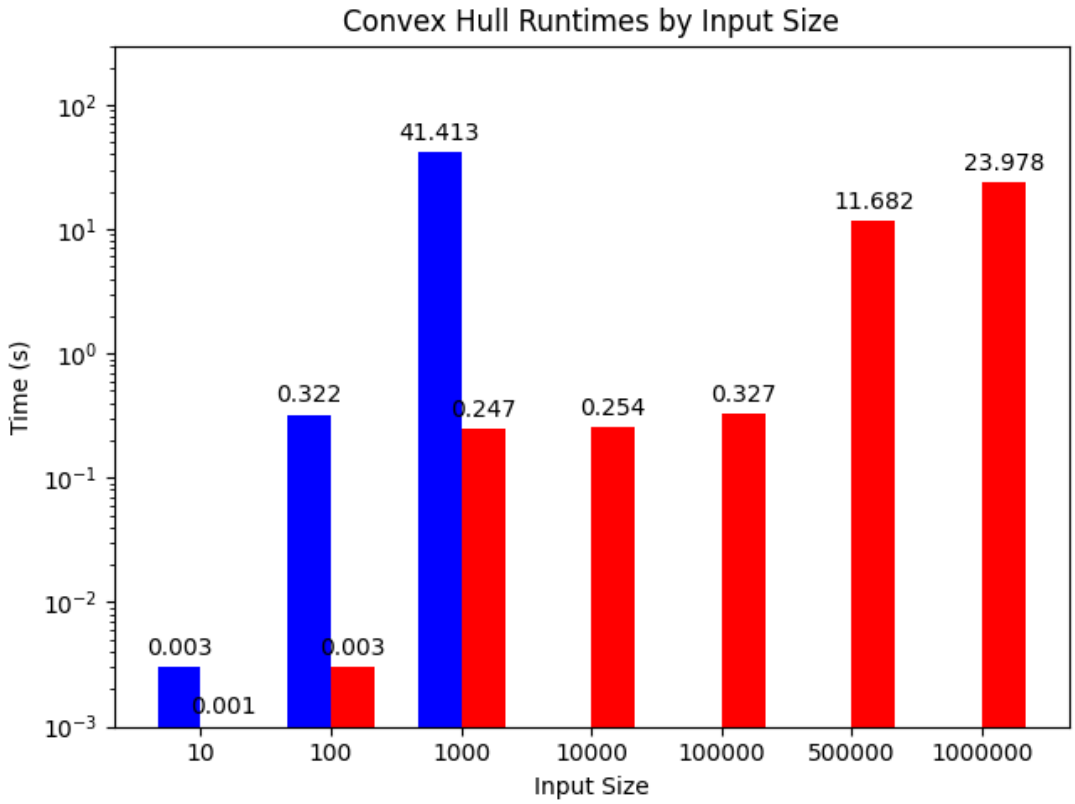
Caroline Pasyanos
CSCI 716 - Computational Geometry
Assignment 1

| Runtime Analysis | | |
|---|---|---|
| **Input Size** | **Brute Force Runtime** | **Graham Scan Runtime** |
| n = 10 | 00:00.003 | 00:00.001 |
| n = 100 | 00:00.322 | 00:00.003 |
| n = 1,000 | 00:41.413 | 00:00.247 |
| n = 10,000 | > 5 minutes | 00:00.254 |
| n = 100,000 | > 5 minutes | 00:02.327 |
| n = 500,000 | > 5 minutes | 00:11.682 |
| n = 1,000,000 | > 5 minutes | 00:23.978 |




Brute force runtimes are shown in blue, and Graham Scan runtimes are shown in red. Runtimes over 5 minutes were not plotted. The y-axis is shown with a logarithmic scale so that all runtimes are visible.

## Methodology & Analysis

Runtimes listed are the average from running the program three times with the same input sets. Points in each input set were randomly generated. I stopped computation early if the program took more than 5 minutes to run. To generate 500,000 and 1,000,000 points, I had to remove the logic disallowing duplicates because it took too long.

The brute force algorithm took longer than 5 minutes when I tried to run it with an input size of n = 10,000. In contrast, the Graham Scan method took less than 30 seconds, even when testing with 1,000,000 points. This clearly illustrates the importance of an efficient algorithm and the difference between $O(n^3)$ and $O(n \log n)$ efficiency, as well as how quickly a runtime spirals out of control with an $O(n^3)$ runtime.