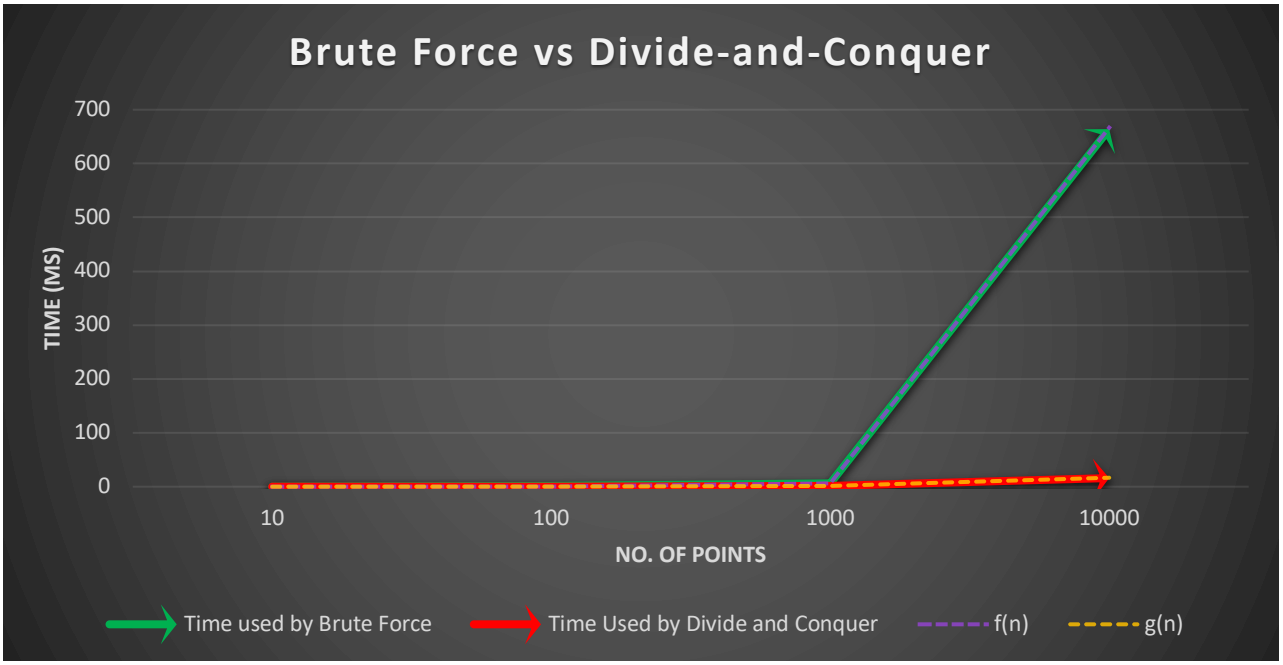
Elijah M.

CS 403

No. of points	Time used by Brute Force	Time Used by Divide and Conquer	f(n)	g(n)	C_1	C_2
10	0	0	0	0.004	*6.67E-06*	*0.000125*
100	1	0	0.07	0.083	6.67E-06	0.000125
1000	6	1	6.67	1.246	6.67E-06	0.000125
10000	664	16	667	16.61	6.67E-06	0.000125



After analyzing my findings, I can conclude, that for $n \leq 100$ both Brute Force and Divide-and-Conquer algorithms perform at a similar rate, while for $n > 100$ Divide-and-Conquer becomes the clear winner in terms of performance and efficiency. By finding the functions $f(n) = c_1n^2$ and $g(n) = c_2nlog(n)$ and comparing them to my algorithms, I can see the connection to their time complexities much clearer. Using my findings, it is easy to conclude that $f(n) = O(n^2)$ and $g(n) = O(nlogn)$. While testing for the values of *c1* and *c2*, I noticed that the constants, *c1* and *c2* have a greater impact the larger *n* gets. With the chosen *c1* and *c2* I was able to find a good

approximation of the time complexities of my algorithms. So we can say *Brute Force = O(n^2)* and *Divide-and-Conquer = O(nlogn)* because by comparing *f(n)* to Brute Force and *g(n)* to Divide-and-Conquer you can draw similarities in how the functions grow at the same rate as the algorithms execution times. In summary, this assignment helped me better understand asymptotic growth rates of functions.