Mod 3 Report

(1) Brute Force Timing

	Time (in seconds)		
Trial #	X = 1000	2X = 2000	3X = 3000
1	0.1150	0.4432	0.9845
2	0.1134	0.4472	0.9939
3	0.1184	0.4795	1.015
4	0.1178	0.4413	1.023
5	0.1175	0.4405	0.9922

I did 5 trials on input sizes of 1000, 2000, and 3000 respectively. As can be seen, the time to compute steadily goes up at the rate expected of an O(n^2) solution which, as we learned in lecture, is the runtime of the naive/brute force runtime of this algorithm and unnecessarily fast. The data supports what we have learned.

(2) Proving Divide and Conquer Solution is o(n^2) and Better than Brute Force

	Time (in seconds)		
Trial #	X = 1000	2X = 2000	3X = 3000
1	0.0103	0.0157	0.0240
2	0.0086	0.0182	0.0228
3	0.0085	0.0169	0.0252
4	0.0095	0.0161	0.0237
5	0.0098	0.0185	0.0239

Clearly, instead of the fairly quick increase in computing time as input size went up in the brute force solution, the time to compute each of these goes up much more slowly. This is exactly what we were looking for and demonstrates very well that the divide and conquer algorithm computes in o(n^2) as opposed to the Theta(n^2) performed by the brute force method. These slowly increasing times also prove that the strip was handeled properly.