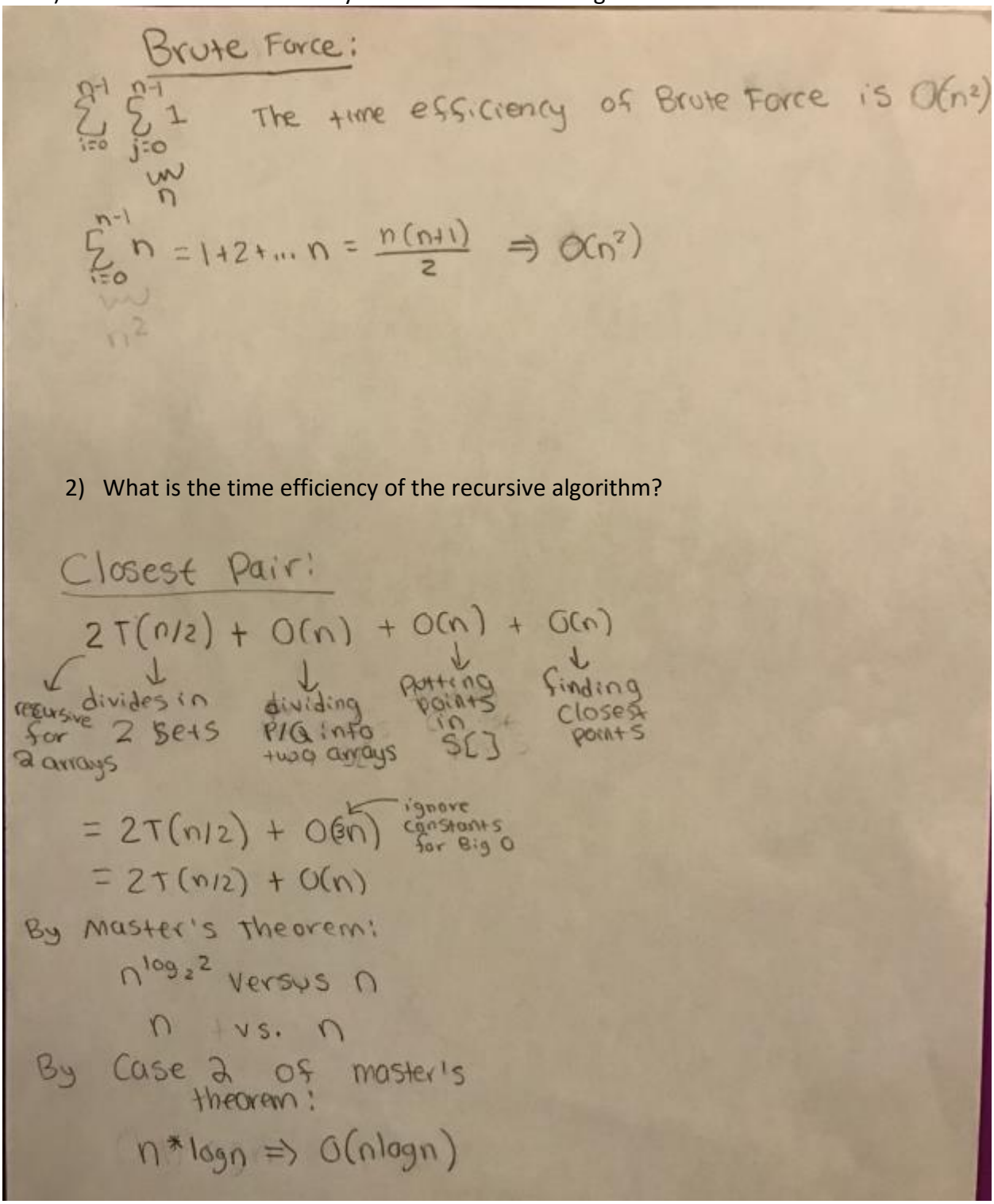
Executive Summary

I first conducted my experiment by making sure my two algorithms, Brute Force and Closest Pair, are running. Then I created a main function so that when you click run, the program will automatically produce the smallest distance between the points for each of the two algorithms. I had the function return the answer to make sure the algorithm is correct and the time to compare time efficiencies (the work is shown on the following page). Some of the lessons I learned were that there is always more than one way to approach a problem and each way can be more efficient than the other. When comparing the times it took for each algorithm to run based on the time function I wrote, I saw that Closest_Pair, the recursive function, often took the least amount of time to produce the result as opposed to the Brute Force. This is because it uses a divide and conquer approach to the problem. First it takes in the points, divides them into two arrays, recursively calls the function again, and then finally merges the two arrays and produces the answer. However, Brute Force, the non-recursive algorithm, just calculates all the distances between the points and returns the smallest one. While it does return the right answer, the recursive function is more time efficient.

Points	Answer	Brute Force (time)	Closest Pair (time)
[(0, 0), (7, 6), (2, 20), (12, 5), (16, 16), (5, 8), (19, 7), (14, 22), (8, 19), (7, 29), (10, 11), (1, 13)]	2.828	0.000534	0.000201
[(0,1),(99,50),(153,22),(44,11)]	45.122	0.000065	0.000095
[(8,0), (0, 6), (0, 0), (10, 3), (4, 20), (2, 11), (15, 9), (7, 0), (19, 11)]	1.000	0.000247	0.000162

1) What is the time efficiency of the non-recursive algorithm?



2) What is the time efficiency of the recursive algorithm?