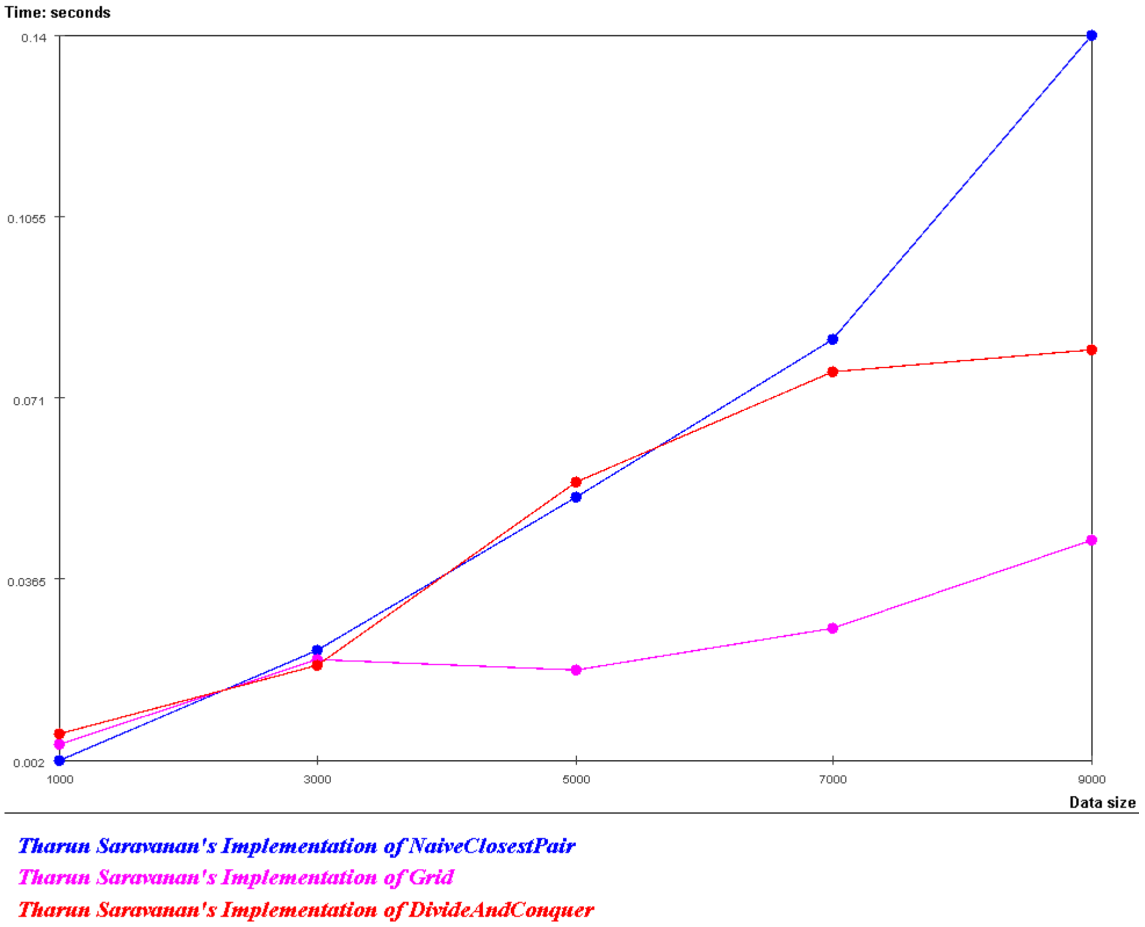
Tharun Saravanan

Professor Simha

Algorithms

17 October 2022

Performance comparison



As shown in the following comparison, while all of the algorithms to find the closest pair begin with similar amounts of time taken to parse the dataset, they drastically differ as the data set gets larger. NaiveClosestPair seems to be the worst algorithm when it comes to finding the closest pair of a large dataset, as the time taken starts to increase at an increasing rate as the dataset gets larger. However, it is still acceptable for sorting a small dataset as shown. DivideAndConquer had a similar runtime to the naive search until the dataset reached a size of 5000. DivideAndConquer was the second best algorithm, but worth noting is that the rate at which divide and conquer seemed to increase similar to a logarithmic function. For this reason, it is possible that DivideAndConquer could actually be the fastest algorithm to find the minimum as the dataset continues to grow. Grid was the overall fastest algorithm with the given dataset, consistently producing a low runtime regardless of the data size tested with.