An Analysis of the Nearest Neighbor Problem

Dong Uk Park 861203574

April 29, 2018

1 Introduction

Based on a set of points given on a plane, the nearest neighbor problem provides the euclidean distance between the closest pair of points.
From the figure above, the two red dots are the closest points based on the entire plane. The equation to find the distance between two points is:

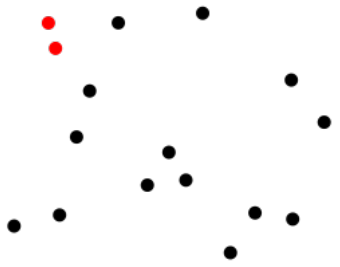
$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

For this assignment, we used two algorithms: Brute Force and Divide and Conquer.

2 Algorithms

2.1 Brute Force Algorithm

For the brute force algorithm, it compared every set of points on the plane and tried to find the lowest distance between the set of points. I used a nested for loop to ensure comparison between every set of points on the plane. A variable min distance tracked the lowest distance between the points.



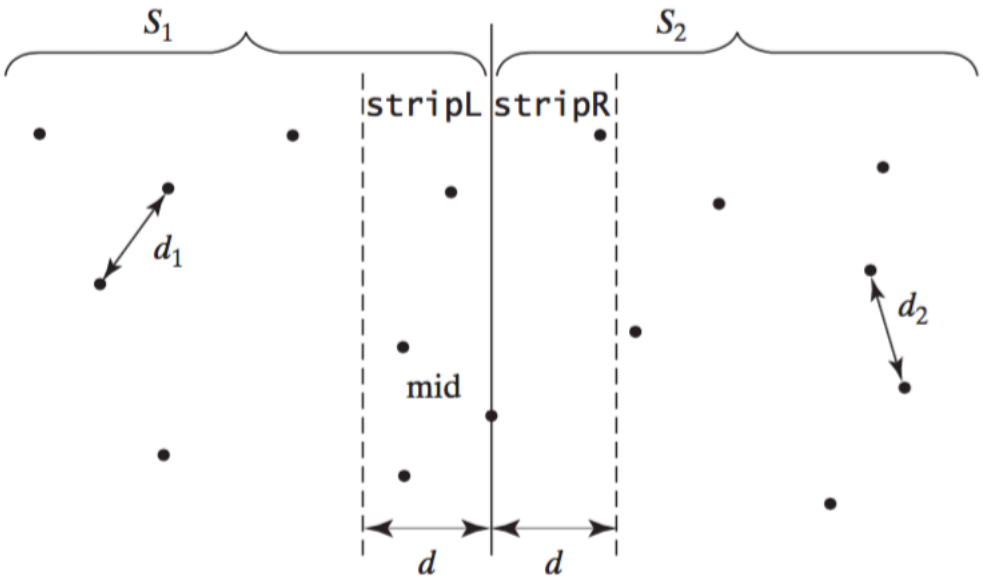
pair.png pair.png

Run-time

The runtime would theoretically be $\theta(n^2)$. This is because of the two for loops going through every single set of points present.

2.2 Divide and Conquer Algorithm

For divide and conquer, I first sorted the points and then called the divide and conquer function. I first divided the array into a left and a right. I recursively called the function until I found the closest pair for the left side. After finding the left side, I did the same for the right side. Based on the disrtances, I compared the two values and chose the side with the smallest distance. After discarding points either less than the midpoint - the smallest distance or greater than the midpoint + the smallest distance. Then, going through the remaining set of points, found the pair that minimizes the smallest distance.



From the figure above, d1 nd d2 are the smallest distance for the left and right side. Then from the mid section, I try to find the closest pair within that region after getting rid of the rest of the points.

Run-time

The theoretical runtime would be $\theta(n \log n)$.

3 Results

	Brute force (sec)	Divide and conquer (sec)
input10	0.0002799	0.0002715
input100	0.0059912	0.0106050
input1000	0.5912289	1.1180930
input10000	59.149109	113.58174
input100000	30 min +	30 min +
input1000000	Never finished	Never finished