

Programming Assignment 1

Nearest Neighbor

Due: April 24, 2015 @ 11:59pm

Problem Description

Input: A set of points in the plane, $\{p_1 = (x_1, y_1), p_2 = (x_2, y_2), \dots, p_n = (x_n, y_n)\}$

Output: The distance between the closest pair of points: that is, the pair $p_i \neq p_j$ for which the distance between p_i and p_j , that is,

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

is minimized.

Divide-and-Conquer

Here's a high-level overview of the divide-and-conquer algorithm:

- Find a value x for which exactly half the points have $x_i < x$, and half have $x_i > x$. On this basis, split the points into two groups L and R .
- Recursively find the closest pair in L and in R . Say these pairs are $p_L, q_L \in L$ and $p_R, q_R \in R$, with distances d_L and d_R respectively. Let d be the smaller of these two distances.
- It remains to be seen whether there is a point in L and a point in R that are less than distance d apart from each other. To this end, discard all points with $x_i < x - d$ or $x_i > x + d$ and sort the remaining points by their y -coordinate.
- Now, go through this sorted list, and for each point, compute its distance to the subsequent points in the list. Let p_M, q_M be the closest pair found in this way.
- The answer is one of the three pairs $\{p_L, q_L\}, \{p_R, q_R\}, \{p_M, q_M\}$, whichever is closest.

Brute Force

You will also be implementing a brute force version of the algorithm which will compare all pairs of points to find the closest pair. This algorithm will be used for comparison purposes.

Turn-in

- nearest_neighbor.py - *The code that you wrote*
- report.pdf - *Report of the results of your experiments*
- README.txt - *Describing how to run the program*

Your nearest_neighbor.py should be run through command line in the following way:

```
$python nearest_neighbor.py example.txt
```

The output should be a file named <example>_distance.txt with the distance of the closest pair output on one line.

Example: \$python nearest_neighbor.py input.txt

input.txt

```
5.1 8.7  
-1.2 3.7  
-4.5 -6.1  
12.7 14.21  
1.6 3.1  
7.9 15.13  
18.4 -25.3  
11.2 -6.3  
7.1 -3.9  
4.6 2.9
```

input_distance.txt

```
2.86356421266
```

Report

The report needs to contain a run-time analysis of the brute force algorithm vs. the divide-and-conquer algorithm described here. You will need to generate a table that shows the run-time of each algorithm on all of the data sets provided. Then provide a discussion of the theoretical run time, and how it is shown with the generated numbers.