Programming Assignment 1
Nearest Neighbor

Due Friday April 22 at 11:59PM

Problem Description

Input : A set of points in the plane, $\{p_1 = (x_1, y_1), p_2 = (x_2, y_2), ..., p_n = (x_n, y_n)\}$
Output : The distance between the closest pair of points: that is, the pair pi 6= pj 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.

Brute Force

Implement the brute force version of this algorithm which will compare all pairs of points to each other and return the minimum distance.
You will use this algorithm to compare against your divide and conquer solution.

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 n_d the closest pair in L and in R. Say these pairs are $p_{L1}, p_{L2} \in L$ and $p_{R1}, q_{R2} \in R$, 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_{M1}, q_{M2} be the closest pair found in this way.
- The answer is one of the three pairs $\{p_{L1}, q_{L2}\}, \{p_{R1}, q_{R2}\}, \{p_{M1}, q_{M2}\}$, whichever is closest.

Executing Your Program

You must name your program nearest_neighbor.py and read a text file from command line. Your nearest neighbor.py should be run through command line in the following way:

```
$pyhton 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

```
$pyhton nearest_neighbor.py input.txt
```

input.txt	input_distance.txt
5.1 8.7	2.86356421266
-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	

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.

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