

CS 3500 Section A, Spring 2000

Programming Assignment #1 — due Thursday, April 6

General description

For this assignment you will have to implement the divide-and-conquer algorithm described in class for computing the closest pair among n points in the plane (see also Section 35.4 of CLR). The main algorithmic issue that you will need to address is how to implement the “combine” step. Possible approaches are:

- Sort the array Y' by y coordinate in each recursive call. This will lead to $\Theta(n \log^2 n)$ worst case running time overall, but will work well enough in practice.
- Use pre-sorting on y and the “inverse merge” procedure described in the book to pass the y -sorted arrays down the recursion tree.
- Do the y -sorting in the combine step itself, by merging the y -sorted arrays returned by the two “children” recursive calls. This implementation essentially embeds merge sort into the closest pair algorithm.

Specifics — read carefully

1. Instead of Euclidean metric, the distance between two points will be measured in the *Manhattan metric*, defined by

$$\text{dist}(p, q) = |p_x - q_x| + |p_y - q_y|$$

2. Input points will be in *general position*, i.e., no two points will share the same x or y coordinate.
3. Input points will be given in increasing order of their x coordinate.
4. Input points will have integer coordinates (between 0 and 10,000,000). Hence, the Manhattan distance between any two input points will also be integer.

Under conditions 1 and 2 above, how many array positions following each point in the y -sorted Y' need to be checked in order to make sure the combine step is correct?

Programming language

You may use either C, C++, or Java for completing this project. Using a different language may be possible, but only with permission from instructor.

Input format

Your program should read the data from the standard input. The input will consist of the number of points, n , on a line by itself, followed by n lines containing, in this order, the x and y coordinates of the n points. As specified above, points are given in increasing order of their x coordinate. Test files (`s10`, . . . , `s100000`) are available in `im7/Pub/3500` on acme. Avoid copying these files in your own directory—run your program with a command similar to “`myprog < ~im7/Pub/3500/s10`” instead.

Output

Your program should output the distance between the points in the closest pair, as an integer.

Turn-in

E-mail your source code to `mandoiu@cc` by midnight, Thursday, April 6. Also include a brief description of your implementation choices, in particular of your method of implementing the combine step. Your code should compile and execute OK on acme.

Grading

90% correctness, 10% programming style. No credit will be given if you implement the naive $\Theta(n^2)$ algorithm instead of the divide-and-conquer algorithm.