Dept. of Computer Science
Wesley W. Terpstra
Dr.-Ing. Ilia Petrov
Prof. Alejandro Buchmann, Ph. D.
Summer Term 2009

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction to Computer Science II

5. Lab

Thanks to our amazing success, due in large part to your hard work, we have managed to greatly reduce the threats to our great nation. Unfortunately, due to the economic crisis, the government has decided to reward our hard work by cutting our funding. Given improved security, they believe that we no longer need so many resources. Despite the distasteful nature of this assignment, it is our duty to comply.

Accounting has identified our undercover agents as a cut-back cash cow. We have nearly 10000 agents placed throughout the world, and all of them are earning danger pay. In order to have a minimal impact on our operational coverage, it has been decided to lay off agents who are near other agents.

Every agent has a home address, which for security reasons is provided to you as (x, y) coordinates. The curvature of the Earth has been eliminated from the input data. Your task is to find two agents who have minimal distance between them. Distance is defined with respect to the Euclidean metric $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

While it is clear that you could simply compute the distance of every pair of agents, this would take too long for 10000 agents. Your experience suggests that one can use sorting in two dimensions to solve this computational geometry problem. Combined with a divide and conquer algorithm, you believe an $O(n \log n)$ algorithm can be realized.

It seems quite likely that this problem has been solved before. Being a resourceful computer scientist, you feel that reading before writing might be a good approach.

As always, use your undercover contacts in the GDI2 lecture to discuss the problem, but for security make sure you fully understand the solution and implement it yourself.

You must implement: Layoffs.java to identify the agent to fire!

The bureau needs this assignment completed by **23:59 July 5th, 2009**!

Handle carefully as this assignment is prone to self destruct!

Step 0: Download the skeleton code from `https://www.dvs.tu-darmstadt.de/teaching/inf2/2009-en/Problem5.zip`

Step 1: Implement sorting of agent coordinates by X and Y.

Hint: Read the documentation about java Comparators.

Step 2: Divide a list into two lists on either side of a vertical line.

Note: The order of coordinates in the output lists must remain the same as they appeared in the input list.

Step 3: Filter a list to keep only those agents within $\pm$ width of the vertical line.

Note: The input and output lists must remain sorted by Y the axis.

Step 4: Implement a method which finds the two closest agents within 7 steps in a list.

Hint: Using a fixed-length inner loop still has linear time complexity.

Step 5: Find the closest two agents.

Hint: Presort the list in findClosest(agents) for recursive use by findClosest(X, Y).

Hint: Read the additional information linked below!

Hint: Divide and conquer around a vertical line. As part of combining your results, look for agents who are close to each other on either side of this line.

Food for thought: Why is checking only the following 7 vertically sorted agents enough?

Information about this algorithm can be found in

a) Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. *Introduction to Algorithms*, 2nd edition. Chapter 33.4.

b) `http://en.wikipedia.org/w/index.php?title=Closest_pair_of_points_problem&oldid=288442793`