

Documentation

Suppose that there are n laptops each containing a wireless transmitter. For each laptop i , following information are known:

- Position, (x_i, y_i) ,
- Wireless transmission range, r_i

That is, we can imagine that the wireless range of laptop i is a circle centered at (x_i, y_i) with radius r_i . We say that the laptops i and j can communicate if their wireless ranges overlap. Of course, not every laptop can communicate with every other laptop, but laptops can send messages by using intermediate laptops as routers. Hop distance $h(i, j)$ is defined as the minimum number of intermediate laptops used to send a message from laptop i to laptop j . For example, if two laptops can communicate directly, the hop distance between them is 1. Now, you are asked to do the following:

- Given a set of n agents with their positions and wireless ranges, design an algorithm based on **Breadth-first Search** (BFS) to compute the hop-distance from the first laptop to every other reachable laptop(s). If agent i is not reachable from agent j then the hop distance $h(i, j)$ will be set to 0. You need to have a pdf document file, **report.pdf**. In this document, give a step by step verbal description of your algorithm in detail.
- Give time and space complexity of your algorithm.
- Implement your algorithm in Java. Input file name should be given as command line argument.
Input file format should be as follows. The first line will contain an integer n , which is the number of agents. What follows will be the n lines, where the i 'th line contains three real numbers x_i , y_i and r_i corresponding to the x and y coordinate of the i 'th laptop, along with its wireless range. All the values will be separated with tabs (`\t`).
Output file format should be as follows. There will be n lines. On the i 'th line will be an integer, representing the hop distance $h(1, i)$, the minimum number of intermediate laptops necessary for the first agent to communicate with agent i .
