/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* BADS 2012 assignment 1: GiantBook

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Name1: Mark E. Klitgaard

Email1: mekl@itu.dk

Name2: Frederik T. Hoffmann

Email2: ftho@itu.dk

Group name: Group FRV

Operating system: Windows 7, Ultimate Edition

Compiler: javac

Text editor / IDE: Notepad++

Optional: Total hours to complete assignment: Unknown.

Please mark one of the following boxes with an X

[X] Yes, to the best of our knowledge, our code works as it should.

[ ] No, our solution does not work. (We will not get credit for this.)

Here's what doesn't work:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Simulation results

\*

\* Fill out the following table with the results of your simulation.

\* I've entered my own results for the first line; you need to

\* replace those with your own values and fill out the rest of the

\* table. N is the input size. T is the number of times you have

\* performed every experiment.

\*

\* giant is the average number of rounds

\* needed until the giant component emerges.

\* nosingles and connect denote the events when the last individual

\* becomes connected, and when the entire network is connected.

\* (The standard deviation is given in parentheses after each

\* average).

\* You probably don't have time to run T=100 experiments for

\* N=10,000,000.

\* If your code isn't fast enough to produce a result, write "n/a" in

\* the corresponding entry.

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

N T giant (stddev) nosingles (stddev) connect (stddev)

--------------------------------------------------------------------

*100: 100 70.31 5.138 250.58 57.256 250.58 57.256*

*1000: 100 694.63 17.468 3895.55 826.27199 3895.55 826.27199*

*10000: 100 6944.47 59.148 48855.72 6501.123 48855.72 6501.123*

*100000: 100 69290.84 185.114 610242.73 68435.217 610242.73 68435.217*

*1000000: 100 693145.9 564.660 7311951.92 82944.858 7311951.92 82944.858*

*10000000: 100 6931262.89 1633.043 8.311E7 6010553.762 8.311E7 6010553.762*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Algorithm analysis

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Assuming we never run out of memory or heap space, if we let our

algorithm for detecting the emergence of a giant component run for 24

hours, it could compute the answer for

*N = 1.600.176.833.280*

*By testing when a giant component emerges on different populations (100,500, 10000, 1000000), we find the function f(x) = mx+b, linear, where x is the elements and f(x) is the time it takes to find a giant component.*

We've run the code using a quick-find implementation as well.

In 1 hour, the largest instance we could manage had

*N = 11291,391*

*We found that quick-find is a power function, and through the equation for our function we calculated the N-value*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Some network science

\*

\* Here we pretend to do a bit of network science using our system.

\* We're a few decades too late to get our results published.

\* Still, it's a valid example that illustrates how algorithms and

\* computers are used in the (other) sciences.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

The occurence of a giant component happens at a time

[X] linear in the size of the network

[ ] quadratic in the size of the network

[ ] logarithmic in the size of the network

[ ] something else, maybe ...

The whole network is connected at a time

[X] linear in the size of the network

[ ] quadratic in the size of the network

[ ] logarithmic in the size of the network

[ ] something else, maybe ...

The last isolated individual becomes connected

[ ] around the time that the giant component emerges

[X] around the time that the whole network becomes connected

[ ] something else, maybe ...

Finally, was it important that I defined the giant component to be of

size N/2, or could I have used N/10? 9N/10? How did you find your

answer (thinking about it? making an experiment? reading an article?)

*By experimenting, we can see that the number of rounds it takes until a giant component emerges depend on how we define a giant component. It makes sense, that if we set giant = N/10, then it will emerge faster than giant = N/2, as it only requires 10% of the network to be connected instead of 50%.*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* List whatever help (if any) that you received, including help

\* from TAs or fellow students. (Such help is allowed, but we want

\* you to acknowledge it.)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* List any other comments here. Feel free to provide any feedback

\* on how much you learned from doing the assignment, and whether

\* you enjoyed doing it. In particular, tell us how this exercise

\* could be improved.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/



