Project 2

The subject of this project is to create an implementation of the Nagamochi-Ibaraki algorithm (see in the lecture notes) for finding a minimum cut in an undirected graph, and experiment with it.

Tasks:

1. Explain how your implementation of the algorithm works. Provide pseudo code for the description, with sufficient comments to make it readable and understandable by a human.

2. Write a computer program that implements the algorithm. You may use any programming language under any operating system, this is entirely of your choice.

3. Run the program on randomly generated examples. Let the number of nodes be fixed at $n = 21$, while the number m of edges will vary between 20 and 200, in steps of 4. For any such value of m, the program creates 5 graphs with $n = 21$ nodes and m edges. The actual edges are selected randomly. Parallel edges and self-loops are *not* allowed in the original graph generation. Note, however, that the Nagamochi-Ibaraki algorithm allows parallel edges in its internal working, as they may arise due to the merging of nodes.

4. Experiment with your random graph examples to find an experimental connection between the number of edges in the graph and its connectivity $\lambda(G)$. (If the graph happens to be disconnected, then take $\lambda(G) = 0$.) Show the found relationship graphically in a diagram, exhibiting $\lambda(G)$ as a function of m, while keeping $n = 21$ fixed. Since the edges are selected randomly, therefore, we reduce the effect of randomness in the following way: run 5 independent experiments for every value of m, one with each generated example for this m, and average out the results. (This is why 5 graphs were created for every m.)

5. For every connectivity value $\lambda = \lambda(G)$ that occurred in the experiments, record the largest and smallest number of edges with which this

λ value occurred. Let us call their difference the *stability* of λ, and let us denote it by $s(\lambda)$. Show in a diagram how $s(\lambda)$ depends on λ, based on your experiments.

6. Give a short explanation why the functions found in items 4 and 5 look the way they look. In other words, try to argue that they indeed show the behavior that can be intuitively expected, so your program is likely to work correctly. Note that it is part of your task to find out what behavior can be expected.

7. Also include a section in the project document that is often referred to in a software package as "ReadMe file." The ReadMe file (or section) provides instructions on how to run the program.

Important note: If there is anything that is not specified in this project description, that automatically means it is left to your choice.

Submission guidelines

Describe everything, including algorithms, program, sources, results and figures neatly and clearly in a study. Include everything in a *single document* that can be read as a report. It should have a *professional appearance,* scanned handwriting, hand-drawn figures etc. are not acceptable! The preferred file type is pdf. Do not submit executable code, but include the source code as an Appendix in the document.

Submit the document through eLearning. Do not send it via e-mail!

Notes:

- The work should be fully individual and original. Any form of cheating is a serious violation of University policies and can lead to serious consequences. Also note that while there were *similar* projects in earlier semesters, the *exact same* project has never been assigned. The minor differences between this and earlier similar projects make it easy to detect if a submission is copied from an earlier one.

- It may be helpful to think about the whole project presentation that your task is not only to solve a technical problem, but you also have to "sell" the results. Try to look at your work from the viewpoint of a potential customer, to whom you want to sell such a software product. How convincing would your presentation look for a customer?

Evaluation

The evaluation will focus on how well each of the specific tasks have been carried out, as well as how professional the whole presentation looks. Projects will be graded on a scale of 1...100.