Exercise sheet 12

February 18, 2020

In this exercise you will learn how to extend the class *graph* from the lecture by a set of diverse methods. For a detailed specification of the methods, please have a look at the file *Graph.H* on the homepage.

Exercise 1 *(5 points)*
Implement the method *readFromFile*, which reads the directional graphs from a file. For specification of the data format, please have a look at the file *Graph.H* on the homepage.

Exercise 2 *(5 points)*
Implement the method *computeReachableNodes*, which marks all reachable nodes from a given node. Apply *breadth first search* or *depth first search* as explained in the lecture.

Exercise 3 *(5 points)*
Implement the method *computeLargestConnectedComponent* which calculates all connected components and marks the nodes in the biggest connected component (= those with the most nodes).

Exercise 4 *(5 points)*
Write a script *LargestConnectedComponentMain*, which reads the linked street graph *saarland.graph* on the homepage and calculates the biggest connected component. All one-way streets are already saved there as two directed edges. Fill in your results in the table linked on the homepage and specify: number of nodes in the original graph, number of nodes in the biggest connected component, and overall running time of your script.
Hint:
You have to enhance the stack size in Java to ensure the function of the recursive calls. Therefore start your script with:

```
java -jar -Xss1024k LargestConnectedComponentMain.jar
```

Commit

Commit your code into the SVN in a new subdirectory **uebungsblatt_12** and a PDF with the solutions of the theoretical tasks in the same folder. Commit your feedback in a text file *erfahrungen.txt* as usual. Please specify: The length of time needed for the exercise. Which tasks have been difficult for you and where did you have problems? How much time did you spend to solve the problems?