Assignment 10

**Valeria Fionda, Mouna Kacimi,
Werner Nutt, Simon Razniewski**

Algorithms on Graphs

Instructions: Your assignment should represent your own effort. However, you are not expected to work alone. It is fine to discuss the exercises and try to find solutions together, but each student shall write down and submit his/her solutions separately. It is good academic standard to acknowledge collaborators, so if you worked together with other students, please list their names.
Be prepared to present your solution at the lab. If you are not able to explain your solution, this will be considered as if you had not done your work at all.
You can write up your answers by hand (provided your handwriting is legible) or use a word processing system like Latex or Word. Experience shows that Word is in general difficult to use for this kind of task.
For a programming task, your solution must contain (i) an explanation of your solution to the problem, (ii) the Java code, in a form that we can run it, (iii) instructions how to run it. Also put the source code into your solution document. For all programming tasks, it is not allowed to use any external libraries ("import") or advanced built-in API functions (for example, `String.indexof("a")` or `String.substring(1,5)`), if not stated otherwise.
Please, include name, matriculation number and email address in your submission.

1. Communication Network

A wireless network can be modelled as a graph where the vertices are computing devices located in physical space. The graph contains an edge (u, v) if device v is close enough to device u to receive and to transmit signals. Two devices are connected unreliably, if the connection can be destroyed by removing just one device from the network.

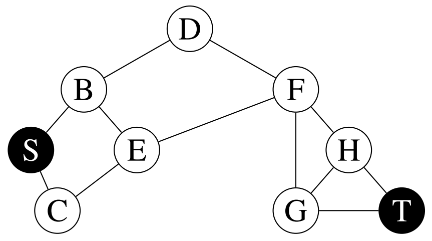

Figure 1: Graph representing a wireless network.

Task 1. In Figure 1 identify such vertices that, if deleted, destroy all paths from S to T.

Task 2. Write an algorithm in pseudo code and Java that given a communication network and two nodes u and v checks if u and v are connected unreliably. Which is the complexity of your algorithm?

(15 Points)

2. Directed Acyclic Graphs

Installed software packages in some Linux distribution form a directed acyclic graph where edge (A,B) means that package A depends on package B. When package B is removed, all dependent package should also be removed. When package A is removed and package A was the only package that depends on B, package B should also be removed.

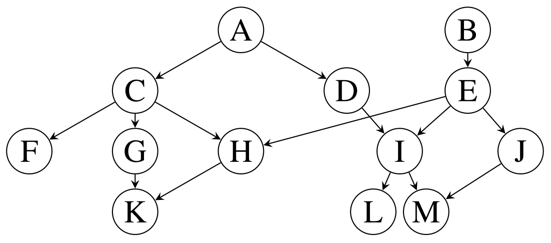

Figure 2: Package dependency graph

Task 3. List the packages that are deleted when

a) package B is deleted

b) package C is deleted

Task 4. Write an algorithm in Pseudocode and Java that given a package dependency network allows for the package deletion. Analyse your algorithm and estimate the asymptotic complexity.

(15 Points)

Remarks: You are allowed to use predefined Java structures such as java.util.LinkedList and java.util.Hashtable.

Submission: Thu, 31 May 2012, 8:30 am. Preferably by email.

If you want to submit a hand-written solution, hand it over to your lab tutor or in the lecture. However, all code has to be submitted in electronic form.