Advanced Software Technology — WT 2016/17

Prof. Gerhard Kraetzschmar — Exercise Sheet 7

Distributed on Oct-13 — Due Date: Nov-13

30 Exercise 13: Graphs and Topological Sorting

Given is a directed graph $G = \langle N, E \rangle$, where $N = \{n_i\}$ is a set of nodes (or vertices) and $E = \{(n_i, n_j) \mid n_i, n_j \in N\} \subseteq N \times N$ is a set of directed edges or links between nodes. The Graph is given by virtue of its adjacency matrix $A = \{a_{ij} \mid a_{ij} = 1 \iff (n_i, n_j) \in E\}$ and 0 otherwise.

1. Define Java classes to represent graphs and adjacency matrices.
2. Write a Java class that can read and write adjacency matrices to and from files.
3. Write a Java class that takes an adjacency matrix and produces the respective graph, and vice versa.
4. Extend the adjacency matrix class by methods that allow to determine whether the graph is directed or undirected and whether it is acyclic or not.
5. Write a Java class that determines the topological depth TD and produces a topological sort TS of the graph G as follows:
   - The topological sort $TS = \{S_k \subseteq N \mid \forall i, j : i \neq j \implies S_i \cap S_j = \varnothing\}$ is a decomposition of the set of graph nodes such that $\forall (n_p, n_q) \in E : \exists r, s : n_p \in S_r \wedge n_q \in S_s \wedge s > r\}$.
   - The topological depth $TD = |TS|$ of a Graph G is the number of sets in the topological sort TS.
6. Test your programs with various examples.

30 Exercise 14: Shortest Paths in Graphs

Given is an undirected graph $G = \langle N, E \rangle$ as in the previous exercise and two nodes $n_s, n_g \in N$.

- Write a Java class with a method that takes a graph and two of its nodes as input, finds the shortest path between n_s and n_g, and produces a traversal list $TL = \langle n_s, \ldots n_g \rangle$ of nodes that need to be traversed to get from n_s to n_g.

30 Exercise 15: Shortest Paths in Weighted Graphs

Given is an undirected, weighted graph $WG = \langle N, E \rangle$ where $N = \{n_i\}$ is a set of nodes (or vertices) and $E \subseteq N \times N \times \mathbb{R}$ is a set of weighted undirected edges or weighted links between nodes. The Graph is given by virtue of its adjacency matrix $A = \{a_{ij} \mid a_{ij} = w_{ij} \iff (n_i, n_j, w_{ij}) \in E\}$ and 0 otherwise. The w_{ij} can be interpreted as the cost for traveling from node n_i to node n_j. Furthermore, given are two nodes $n_s, n_g \in N$.

- Write a Java class with a method that takes a graph WG and two of its nodes n_s, n_g as input and finds the lowest cost path between n_s and n_g, and produces a traversal list $TL = \langle n_s, \ldots n_g \rangle$ of nodes that need to be traversed to get from n_s to n_g.