Eidgenössische Technische Hochschule Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Department Informatik
Markus Püschel
Peter Widmayer
Thomas Tschager
Tobias Pröger

1st December 2016

Algorithmen & Datenstrukturen   Exercise Sheet 11   AS 16

Hand-in: Thursday, 8th December 2016 before the start of the lecture at 10:00 in the entrance area of ML D28. Please staple all sheets together and use this sheet as the cover page. Fill out the first two fields of the form below.

Exercise class (Room & Day): ______________________

Submitted by: ______________________

Corrected by: ______________________

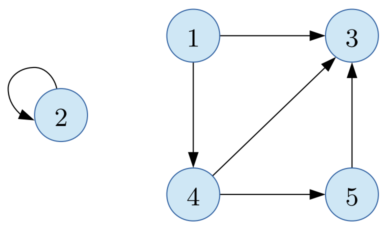
Bonus points: _____

Note: This sheet treats graph theory; we briefly repeat its most important definitions. A *directed graph* $G = (V, E)$ consists of a set V of *vertices* and a set $E \subseteq V \times V$ of *edges*. An *undirected graph* $G = (V, E)$ also consists of a set V of edges and a set E of edges where $E \subseteq \{\{u, v\} \mid u, v \in V\}$. Hence, in directed graphs every edge has a direction and is a pair of vertices while in undirected graphs an edge does not have a direction and is a set of vertices. A bipartite graph $G = (V, E)$ is a graph where the set of vertices can be partitioned into two disjoint sets U and W, such that each edge has exactly one vertex in U and the other one in W. A sequence of vertices $\langle v_1, \ldots, v_k\rangle$ is a *path* if for each $i \in \{1, \ldots, k-1\}$ an edge from v_i to v_{i+1} exists. The length of a path is $k-1$, the number of traversed edges. An undirected graph is *connected* if there exists a path from v to w for every pair of two different vertices v and w. A path $\langle v_1, \ldots, v_k, v_{k+1}\rangle$ is a *cycle* if $v_1 = v_{k+1}$. A cycle of length 1 is called *loop*. A cycle is called *simple cycle* if all contained vertices (with the exception of the first and the last node) are visited exactly once. A graph is called *acyclic* if it does not contain a cycle.

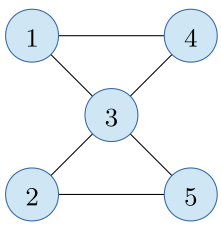
Exercise 11.1 *Properties of example graphs.*

Given the following graphs, respectively adjacency matrices of graphs.

(i)



(ii)



Please turn over.

$$\text{(iii)} \quad \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad \text{(iv)} \quad \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

a) Give the adjacency matrix for the graphs (i) and (ii).

b) Draw the graphs (iii) and (iv).

c) For each of the graphs (i) - (iv) shown above, specify whether they are directed or undirected, whether they are bipartite, whether they are acyclic, and whether they have loops.

d) Additionally, examine for each *undirected* graph of the for graphs given above, whether it is are connected and whether it contains an Eulerian cycle or an Eulerian path. Justify your answer.

e) Compute the reflexive and the transitive closure for the graph (i). Draw the result and write down the corresponding adjacency matrix.

f) Draw an undirected graph with 7 vertices that is not connected, such that exactly two vertices have degree 3 and all other vertices have degree 2. Can you also give an undirected graph with 7 vertices, such that exactly one vertex has degree 3 and all other vertices have degree 2? If yes, draw such a graph, otherwise justify why you cannot draw such a graph.

Exercise 11.2 *Properties of graphs.*

a) Indicate how many edges a directed and an undirected graph with n vertices can have at most and justify your answer.

b) A tree is an undirected graph that is acyclic and connected. Show using mathematical induction over the number of vertices n that every tree has exactly $n-1$ edges.

c) Prove or disprove: Every undirected graph with n vertices and $n-1$ edges is a tree.

Exercise 11.3 *Test for acyclicity.*

a) How can you see from an adjacency matrix if the corresponding graph has a cycle of length 1?

b) Provide an algorithm that takes the adjacency matrix of a directed graph G as input and decides whether G is acyclic or not. Provide the running time of your algorithm.

c) Assume that your algorithm from b) decides that G is not acyclic and that you are additionally given the adjacency matrix of G. How can you reconstruct a cycle? If there exist multiple cycle, it is sufficient to report an *arbitrary* cycle. Provide the running time of your solution.