Exercise VII, Algorithms 2024-2025

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. There are many problems on this set, solve as many as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

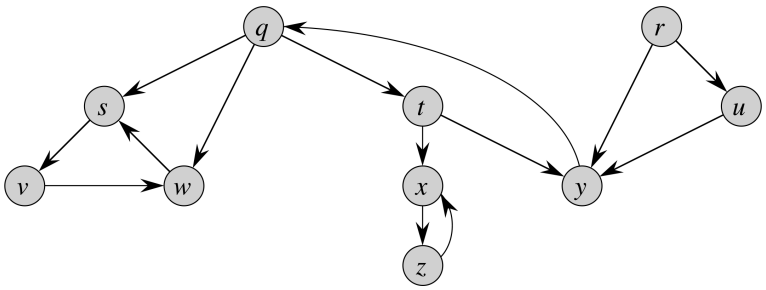
Graph Representation

1 (Adaptation of Exercise 22.1-1) Given an adjacency-list representation of a directed graph, how long does it take to compute the out-degree of a given vertex v? How long does it take to compute the in-degree of a given vertex v?

2 (Exercise 22.1-2) Give an adjacency-list representation for a complete binary tree on 7 vertices. Give an equivalent adjacency-matrix representation. Assume that vertices are numbered from 1 to 7 as in a binary heap.

BFS and DFS

3 (Adaptation of Exercise 22.3-2) Consider the following directed graph:



3a Show how breadth-first search works on the graph if we start with q as a source and each adjacency list is ordered alphabetically. Show the length of the shortest path from the source to each vertex as the algorithm progresses.

3b Show how depth-first search works on the graph. Assume that each adjacency list is ordered alphabetically. Show the discovery and finishing times for each vertex, and show the classification of each edge.

4 (Exercise 22.2-4) What is the running time of BFS if we represent its input graph by an adjacency matrix and modify the algorithm to handle this form of input?

5 (Exercise 22.3-12) Show that we can use a depth-first search of an undirected graph G to identify the connected components of G, and that the depth-first forest contains as many trees as G has connected components. More precisely, show how to modify depth-first search so that it assigns to each vertex v an integer label $v.cc$ between 1 and k, where k is the number of connected components of G, such that $u.cc = v.cc$ if and only if u and v are in the same connected component.