4th Homework Assignment

Spring Semester 2017
Prof. Helmut Alt

Introduction to Algorithms

Due April 3, 2017, 10 a.m.

Exercise 1

7 points

(a) Given an adjacency-list representation of a directed graph, how long does it take to compute the outdegree of every vertex? How long does it take to compute the indegrees?

(b) The *transpose* of a directed graph $G = (V, E)$ is the graph $G^T = (V, E^T)$ where $E^T = \{(v, u) \in V \times V | (u, v) \in E\}$. Describe and analyze efficient algorithms for computing G^T from G for both the adjacency-list and the adjacency-matrix representations of G.

Exercise 2

13 points

Implement directed and undirected graphs using adjacency-lists and adjacency-matrix as the underlying data structure. For this purpose assume that every vertex has a unique identifier which is an integer between 1 and n, the number of vertices.

Assume that for input and output of graphs the following format is used:

There are n adjacency-lists for the vertices with identifiers $1, ..., n$ respectively. The elements in one list are separated by commas, the different lists are terminated by semicolons, e.g.,

2,3;
1,3;
1,2;
;

would describe in the undirected case the complete graph with the 3 vertices 1,2,3 plus one isolated vertex number 4.

Your implementation should be usable for this and further implementations of graph algorithms. For now, the following queries on graphs should be implemented:

(a) Given vertices u, v, answer whether (u, v) (or $\{u, v\}$) is an edge or not.

(b) Given a vertex s give a list of all vertices to which a path from s exists in the order they are visited by depth-first-search with start vertex s.

(c) Given vertices u, v, answer whether there exists a path from u to v or not.

(d) Determine the connected components in the undirected case.