Exercises on Graphs

Giulia Bernardini

May 2022

Exercise 1. Given an adjacency-list representation of a directed graph, how long does it take to compute the out-degree of every vertex? How long does it take to compute the in-degrees?

Exercise 2. 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?

Exercise 3 (Two-coloring)**.** Given a connected, undirected graph with n vertices and m edges, design an algorithm that assigns one of two colors (say blue or green) to each vertex in such a way that no edge links two vertices of the same color; or return FAIL if no such coloring is possible. (**Hint:** Use values $v.distance$ computed during BFS).

Exercise 4. Give an $O(|V|)$-time algorithm that determines whether or not a given undirected graph contains a cycle. (**Hint:** Think of the maximum number of edges that an acyclic undirected graph may have; use DFS and terminate it early when appropriate).

Exercise 5. We are given a directed graph on which each edge (u, v) has an associated value $r(u, v)$, which is a real number in the range $[0, 1]$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices. (**Hint:** modify Dijkstra appropriately.)

Exercise 6. How can we use the output of the Floyd-Warshall algorithm to detect the presence of a negative-weight cycle? (**Hint:** look at the diagonal of the output matrix.)