ECE:5330 Graph Algorithms and Combinatorial Optimization

Spring 2024

Assignment 2

Due date: Feb. 16, 2024

1. Page 67: Problem 3 **(10 points)**
2. Page 68: Problem 6 **(20 (5 + 5 + 10) points)**
3. Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the element in $A[1]$. Then find the second smallest element of A, and exchange it with $A[2]$. Continue in this manner for the first n-1 elements of A. Write pseudo-code for this algorithm, which is known as selection sort. Give the best-case and worst-case running times of selection sort in Θ-notation. **(20 (10 + 5 + 5) points)**
4. Show that a depth-first search of an undirected graph G can be used 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 each vertex v is assigned an integer label $cc[v]$ between 1 and k, where k is the number of connected components of G, such that $cc[u] = cc[v]$ if and only if u and v are in the same connected component. **(15 points)**
5. Page 110: Problem 10 **(15 points)**
6. Page 112: Problem 12 **(20 points)**

Bonus Problem: **(20 (5 + 15) points)**

A communication network, such as the Internet, can be modeled as an undirected graph $G = (V, E)$. Here the vertices V are the machines on the network, and the edge set consists of one edge for each pair of machines that are directly connected. We assume that the edges of G are undirected, that is, if there is a direct connection from machine u to machine v, then there is also a direct connection from machine v to machine u.

It is highly desirable for a communication network graph to be connected, so that every machine on the network can communicate, possibly through a series of relays, with any other machine. But networks can change, with some machines failing and other machines being added to the network. It is useful to have a monitoring algorithm that collects information about the current network graph (vertices and edges) at designated times, and determines properties related to connectivity.

(a) Describe (in words and pseudocode) a monitoring algorithm, which does the following. As input, it is given an undirected graph $G = (V, E)$ representing the current network, in adjacency list format. It should output a Boolean saying whether or not G is connected. Your algorithm should run in time $O(V + E)$.

(b) It is very nice if the network is connected, but even if it is, we might worry that it might become disconnected soon. This is likely if the network graph has *critical vertices*, whose removal would disconnect the graph. That is, u is a critical vertex exactly if there are two other vertices, v and w, for which every connecting path in the graph runs through u (this is equivalent to saying that removing u disconnects the graph, because removing u leaves no path from v to w.)

Describe (in words and pseudocode) a new monitoring algorithm, which does the following. As input, it is given a connected undirected graph $G = (V, E)$ representing the current network, in adjacency list format. It should output a list of all the critical vertices of the graph. Your algorithm should have polynomial running time.