IT251 Lab 5 Problems -- 18 Feb 2020

Instructions:

1. Do the below problems in the order given.
2. For each of the problems below, test your code on different input graphs.
3. Use the adjacency list format in *input.txt* of the previous lab to read in graphs.

Problem 1: Dijkstra's Shortest Path

a) Write a program to implement Dijkstra's shortest path algorithm on a directed graph. Read in the source vertex **s** from the user and the directed graph from the input file *input.txt.* Your program should print the length of the shortest path from **s** to all the vertices in the graph. For each vertex **v**, print the vertices in the actual shortest path from **s** to **v.**

b) There can be more than one shortest path to a given vertex **v** from **s.** Modify the above program to additionally compute if there is only one unique shortest path from **s** to **v**. For every vertex print **TRUE** if the shortest path from **s** to that vertex is unique, **FALSE** if not.

Problem 2: Computing SCCs

Write a program to compute the SCCs of a directed graph. Read in a directed graph from the input file *input.txt.* Print the vertices in each SCC of the graph.

Problem 3 [DPV 3.16]:

Suppose a CS curriculum consists of n courses, all of them mandatory. The prerequisite graph G has a node for each course, and an edge from course v to course w if and only if v is a prerequisite for w. Find an algorithm that works directly with this graph representation, and computes the minimum number of semesters necessary to complete the curriculum (assume that a student can take any number of courses in one semester). The running time of your algorithm should be linear. Read in the (directed) prerequisite graph from the input file *input.txt.*