University of Houston

# COSC 3320: Algorithms and Data Structures
# Spring 2016

## Homework 7

Due April 28, at the start of class

1. Given two strings $X$ and $Y$, a third string $Z$ is a *common superstring* of $X$ and $Y$ if $X$ and $Y$ are both subsequences of $Z$. (Example: if $X =$ `sos` and $Y =$ `soft`, then $Z =$ `sosft` is a common superstring of $X$ and $Y$.) Design a dynamic programming algorithm which, given as input two strings $X$ and $Y$, returns the length of the shortest common superstring (SCS) of $X$ and $Y$. Specifically, you have to write a recurrence relation $\ell(i, j) = |SCS(X_i, Y_j)|$ that defines the length of a shortest common superstring of $X_i$ and $Y_j$, and the pseudocode. The algorithm, which has to return $\ell(n, m)$, must run in time $\Theta(n \cdot m)$, where $n = |X|$ and $m = |Y|$. (Hint: use an approach similar to the one used to compute the length of a LCS of two strings.)

2. Consider the following simple graph, represented by its adjacency matrix.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

   (a) Draw the graph.

   (b) Run the DFS algorithm starting from vertex 1, and draw the final DFS tree.

   (c) Run the BFS algorithm starting from vertex 1, and draw the final BFS tree.

3. Let $G = (V, E)$ be a graph with $n$ vertices and $m$ edges. Design and analyze an algorithm that returns, if it exists, a vertex $i \in V$ such that at least $n/2$ different vertices are reachable, via a path, from $i$. (Hint: Use the BFS algorithm.)

4. Consider the following weighted graph, represented by its adjacency matrix.

$$\begin{bmatrix} 0 & 3 & 0 & 0 & 0 & 4 & 1 \\ 3 & 0 & 10 & 0 & 0 & 0 & 4 \\ 0 & 10 & 0 & 7 & 0 & 0 & 8 \\ 0 & 0 & 7 & 0 & 6 & 0 & 5 \\ 0 & 0 & 0 & 6 & 0 & 5 & 4 \\ 4 & 0 & 0 & 0 & 5 & 0 & 2 \\ 1 & 4 & 8 & 5 & 4 & 2 & 0 \end{bmatrix}$$

   List the edges of the minimum spanning tree in the order they are added by Kruskal's algorithm.