

Data Structures and Algorithms

Assignment #6

Due: Friday March 7

1. (20 points).

Recall the Maze program in Section 8.7 and Lecture 16 Slide 17.

Use disjoint union/find to design an algorithm to design random mazes which have the property that there is no path from start to end. As in the lecture, the maze must have no cycles and from each cell there is a path from it to one of start or end. You can assume that the cells are numbered 1 to n , with 1 being the start cell and n being the end cell. The edge set E contains all possible pairs of edges in the maze not including the boundary edges.

Repeat the same problem but now add the constraint that the removal of a **pre-specified wall** (i.e., a pre-specified edge) will create a unique path from start to end.

2. (5 points).

Give an algorithm that detects whether a directed graph has a cycle. Use the adjacency list representation for graphs. (Of course you can have additional field(s) for each node in addition to their name and pointers to successors.)

3. (10 points)

Give an algorithm that detects whether a directed graph is a forest. Use the adjacency list representation for graphs. (Of course you can have additional field(s) for each node in addition to their name and pointers to successors.)

4. (No points but start working on it).

In preparation for your next programming assignment you should read Weiss Chapter 10 Section 10.1.2. on Huffman Codes. Start thinking (and even programming) on how to read a paragraph of text and recording the frequency of each character. The characters will be small and capital letters, digits, punctuation marks such as ".", ",", and ";", blanks (spaces between words), and new line (carriage returns). You can disregard the latter (new line/carriage returns) and encode them as blanks.

The next step will be to implement Huffman optimal algorithm. Brush up on how to implement binary heaps!