

CSE 326 DATA STRUCTURES

HOMEWORK 5

Due: **Friday, November 9, 2007** at the beginning of class.

Problem 1. Unions

Show the result of the following sequence of instructions:

```
union(1,2), union(3,4), union(3,5), union(1,7), union(3,6), union(8,9),
union(1,8), union(3,10), union(3,11), union(3,12), union(3,13), union(14,15),
union(16,0), union(14,16), union(1,3), union(1,14)
```

Use figures such as those on page 296 of Weiss. As usual, showing some steps will allow for the possibility of partial credit. If the two trees have the same height (b) or size (c), make the larger key the root.

- (a) Perform all unions arbitrarily (The RIGHT tree, or equivalently, the root with larger key, is the root of the new tree).
- (b) Perform all unions by height.
- (c) Perform all unions by size.

Problem 2. The Golden Gate

Design an algorithm that generates a maze that contains no path from start to finish, but has the property that the removal of a *prespecified* wall creates a unique path. (Hint: You may use the pseudocode given to you in the slides from class. If you do, you'll only need to add roughly two lines of pseudocode and change one existing line.)

Note that you can, if you wish, wait until after your algorithm has completed to declare which cells are the start and end cells. If you choose to follow this idea, you don't need specify the selection of start and end cells in your algorithm, but explain how they could be selected after your pseudocode.

Problem 3. Deunion

Suppose we want to add an extra operation, **deunion**, which undoes the last **union** operation that has not been already undone.

- (a) Show that if we do union-by-height and **finds** without path compression, then **deunion** is easy and a sequence of M **union**, **find**, and **deunion** operations takes $O(M \log N)$ time.
- (b) Why does path compression make **deunion** hard?