CSE 326: Data Structures
Practice Problems/Suggestions for Final
June 3, 1999

Global Comments:

- Material to be covered: comprehensive! That means Weiss, Chapters 2– 5, 6.1– 6.4, 7.5, 8, 9.1– 9.3, 9.5, 9.6.1, plus handout on splay trees.
- Review session: Monday, June 7, 5pm, EE 1 037.
- The exam will be closed books, closed notes, no calculators.
- Be sure you know:
  - how to analyze the running times of simple programs, notions of worst-case and average-case complexity
  - how to compare functions by Big Oh, Big Omega and Big Theta.
  - tree traversals (preorder, inorder, postorder)
  - how the basic dictionary operations are implemented, what their worst-case (and when appropriate, average case) running times are and what the tradeoffs are using: unsorted and sorted lists, binary search trees (without balancing), AVL trees, splay trees, B-trees, hashing (with separate chaining or the various open addressing)
  - how to use heaps to implement a Priority Queue ADT
  - Heapsort
  - graphs
    * basic definitions and properties
    * depth-first and breadth-first search
    * topological sorting
    * Dijstra's shortest path algorithm
    * Kruskal's minimum spanning tree algorithm
  - how the basic Disjoint Set ADT operations are implemented (Union by size or by height and path compression) and what their running times are.

Practice Problems on Recent Material (not to be turned in):

1. Weiss, p. 379, Problems 9.1, 9.5, 9.7a
2. Weiss, p. 380, Problem 9.10
3. Weiss, p. 381, Problem 9.17, 9.24 (See section 9.6.4)
4. Weiss, p. 382, Problem 9.25 (See section 9.6.4)
5. Let A be the adjacency matrix for a directed graph on n vertices (so all of A's entries are 0 and 1). Give a graph theoretic interpretation for the (i,j) entry of the matrix A^k (the result of multiplying the matrix A by itself k times).
6. Give a lower bound on the number of edges in a connected graph with n vertices.

7. Give an upper bound on the number of edges in an acyclic (not necessarily connected) graph with n vertices.

8. An undirected graph $T = (V, E)$ is a tree if given any two vertices u and v of T, there is a unique simple path from u to v. If T is a tree, then T has the following properties:

   (a) T is connected.

   (b) T is acyclic.

   (c) deleting any edge of T yields a disconnected graph.

   (d) if u, v in V and $e = \{u, v\}$ is not an edge of T, then adding e to T yields a graph with exactly one simple cycle, and that cycle contains e.

   (e) T has exactly $n - 1$ edges, where n is the number of vertices of T.

   For each of these 5 properties, find a graph that has that property but is not a tree. For each of the ten possible pairs of properties, either show that any graph with these two properties is a tree or find a counterexample.

9. The particular topological sort produced by the algorithm we discussed may depend on the order in which the vertices of G are processed by the main procedure and the other in which the neighbors of each vertex are processed. Is it true that *any* topological sort of a DAG can be produced by some depth-first-search?

10. Find a necessary and sufficient set of conditions for a DAG to have a unique topological order.

11. Consider the following algorithm for finding a minimum spanning tree of a connected, undirected graph $G = (V, E)$.

```
Maintain two sets:
   -- a set E' of edges, initially empty.
   -- a set N of vertices that initially contains an arbitrary
           single vertex in V.
Repeat |V|-1 times:
   find an edge e in E of least cost that joins a vertex u in N
   to a vertex v that is not in N and then
       add e to E'
       add v to N.
```

   Prove that the output of this algorithm (V, E') is a minimum spanning tree or give a counterexample.

12. Prove that an undirected graph is bipartite if and only if it has no odd length cycle.

13. Dijkstra's algorithm doesn't work if there are negative weight edges. One proposal for how to deal with this case, is to find the most negative edge, say with value $-v$, add v to the cost of every edge, and solve the single-source shortest path algorithm in the resulting graph. Either prove that this works, or give a counterexample.

14. Weiss, p. 322, Problem 8.1

15. Weiss, p. 323, Problems 8.2, 8.4

16. Consider a sequence of operations that starts with n singleton sets and consists of a sequence of $m \leq n$ Union operations follows by a sequence of $n - m$ Find operations with path compression. Show that the time for the whole sequence is $O(n)$.