ICS 261 – Fall 2003 – Final Exam

Name:

Student ID:

1:

2:

3:

4:

5:

6:

Total:

1. (30 points)

Suppose we use a hash function h to hash n distinct keys into an array T of length m.

(a) Assuming simple uniform hashing, what is the expected number of collisions? More precisely, what is the expected cardinality of $\{\{k, l\} \mid k \neq l$ and $h(k) = h(l)\}$?

(b) For what values of m is the expected number of collisions less than one?

(c) Describe an efficient algorithm for listing all the collisions that would occur for a given set of n keys, choice of hash function h, and choice of m. Your algorithm should use $O(n)$ space even when $m > n$. What is the time complexity of your algorithm, in terms of n and k where k is the total number of collisions for the given input?

2. (15 points)

Suppose we create a binary search tree by inserting a set of n distinct items in random order, where $n > 1$ is an odd number. What is the probability that the median of the set of items ends up at a leaf of the tree?

3. (15 points)

The depth of a node in a tree is defined to be the length of the path from the root to the node. Is it possible to store the depth of each node in a red-black tree, and efficiently update these depths when an item is inserted into or deleted from the red-black tree? Show how, or explain why not.

4. (20 points)

In the *rectangle reporting problem*, the data consists of a collection of rectangles in the plane. In each query, we must list all rectangles that contain a given query point. Describe an efficient data structure for solving the rectangle reporting problem. What are the query time and storage space bounds for your data structure?

5. (20 points)

In the union-find problem, show that any sequence of m make-set, union, and find operations, where all the union operations appear before any of the find operations, takes only $O(m)$ time if both path compression and union by size (i.e. linking the smaller tree to the larger one in each union) are used. What happens in the same situation if only the path compression heuristic is used?

6. (20 points)

In class we method for solving the least common ancestor problem in trees by translating the tree to an instance of the range minimum problem. Suppose we wish to solve least common ancestors in a dynamic tree, subject to the following update operation:

- child(x): make a new tree node that is a child of the existing tree node x.

What is the effect of the child(x) operation on the data for the translated range minimum instance?

You may use this page (or the back of the other pages) as scratch paper.