

Name: _____

October 21, 2004

CS 535 Exam I

Instructions: There are five problems in this exam. Some of them consist of several subproblems. Make sure you look through them carefully. Please write legibly and justify all your answers.

1. Let $T_A(n)$ and $T_B(n)$ be the runtimes of two algorithms that sort n numbers. Suppose both $T_A(n)$ and $T_B(n)$ are $O(n^2)$. Determine whether each statement below is true or false. Please justify your answer with a proof or a counterexample.
 - a. $T_A(n) + T_B(n) = O(n^2)$.
 - b. $T_A(n) - T_B(n) = o(n^2)$ (“little-oh” of n^2).
 - d. $T_A(n) = O(T_B(n))$.
2. Answer three out of the four questions below.
 - a. Describe the location of the item with the second largest key in a binary search tree.
 - b. Show the result of inserting items with keys 2, 1, 4, 5, 9, 3, 6, 7, and then removing the item with key 4 into an initially empty AVL tree. Make sure that you demonstrate each step.
 - c. Compare and contrast separate chaining and linear probing for resolving collisions in hashing.
 - d. Describe how you would construct a skip list for a set of n items whose keys can be ordered.
3. Given a tree T with n nodes, let each node v contain an extra field $count(v)$ – which equals the number of nodes in the subtree rooted at v . For example, $count(r) = n$ where r is the root node. Describe an $O(n)$ algorithm that computes the $count$ fields of all the nodes in T . Why is your algorithm correct?
4. Describe how to perform the operation `removeAllElements(k)` in an ordered dictionary implemented with a binary search tree T , and show that this method runs in time $O(h + s)$, where h is the height of T and s is the size of the iterator returned.
5. The mailing list of a mail-order company is stored as an unsorted linked list, where each item contains a name and an address. To reduce costs, the company wants to remove duplicates from its list. Give an efficient algorithm to perform this task. Note that you can use other types of data structures. You should also do better than the obvious and inefficient $O(n^2)$ -time algorithm that checks all possible pairs of elements, where n is the size of the mailing list.