

Name: _____

March 12, 2009

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.

1. *Asymptotic Notation.*

a. Please rank the following functions by the order of their growth – from slowest to fastest: $4^{\log n}$, $n^2 \log n$, $(1.5)^n$, $\sqrt{4n}$, $2/n$.

b. Please provide brief explanations to your answers for the following questions –

(i) Is $\log n = O(n)$?

(ii) Is $\log n = \Omega(n)$?

(iii) Is $\log n = \Theta(n)$?

2. Heaps are binary trees used to implement priority queues.
- But heaps are not just ordinary binary trees. What are the two important properties of heaps?
 - Let d be some positive integer. A generalization of heaps are d -heaps which behave like ordinary heaps except that each internal node is required to have d children.
- Recall that we used an array A to store a heap. In particular, a node labeled j is stored in $A[j]$. Its parent is labeled $\lfloor j/2 \rfloor$, its left and right children are labeled $2j$ and $2j + 1$ respectively. Suppose we want to store d -heaps in the same manner. Explain how you will label the nodes. If a node is labeled j , what are the labels of its parent and its d children?
- What is the height of a d -heap when it stores n items? Your answer should be in terms of n and d .

3. Short questions. Answer two of three please.
- a. Show the result of inserting items with keys 2, 1, 4, 5, 9, 3, 6, into an initially empty *splay* tree, and then removing the item with key 4. Make sure that you demonstrate each step.
 - b. Compare and contrast AVL trees and splay trees.
 - c. Compare and contrast separate chaining and linear probing for resolving collisions in hash tables.

4. For the ordered dictionary ADT, the following methods are also supported:

`closestKeyBefore(k)`: returns the key of the item with the largest key less than or equal to k .

`closestKeyAfter(k)`: returns the key of the item with the smallest key greater than or equal to k .

Suppose an ordered dictionary is implemented using binary search trees. Describe in pseudocode how each method can be implemented in $O(h)$ time where h is the height of the tree. For this problem, it is okay to assume that there are no duplicate keys in the dictionary.

5. Let S consist of n distinct numbers x_1, x_2, \dots, x_n . Given some value z , we wish to determine if there are two numbers x_i and x_j so that $x_i + x_j = z$. Describe an algorithm that will solve this problem in $O(n \log n)$ time.