COMP 550 Algorithms and Analysis
Mid-Term 2 (SAMPLE)
Spring 2015

Name ____SAMPLE_______________

PID ___________________________

Honor Pledge:
I have not given nor received unauthorized assistance in completing this exam.

Signature ___________________________

Note:

(1) All “lg”s are based 2 if unspecified.

(2) Points of subproblems are evenly distributed within each problem unless specified.

(3) Your score will be put on the second page – to protect your privacy.

1. (6’) True or False:
   (a) _____ The worst case time for CHAINED-HASH-INSERT is O(1).
   (b) _____ Expected time for a search in a chained hash table is O(1).
   (c) _____ Using median finding, Quicksort runs in Θ (n lg n) worst case.
2. (12’) (a) Give an asymptotic upper boundΘ (_______) and lower bound Θ (_______) on the height of a Red-Black Tree having *n* internal nodes.
   (b) A red-black tree has height (number of levels of internal nodes) 4.
   What is the max number of internal nodes in the tree? ___________
   What is the min number of internal nodes in the tree? ___________
3. (18’) Show the red-black trees that result after successively inserting the keys 41, 38, 31, 12, 19, 8 into an initially empty red-black tree. You may double circle the red nodes to differentiate it with black ones. (Hint: Given the style of the question, try not to get it wrong in the early stage – double check the 5 properties after each insertion)

4. (12’) (a) Insert the keys 307, 314, 400, 258, 312, 401, 355 into an initially empty BST.

   (b) Then show the result of a left rotate on 314 in this BST.

5. (10’) Write a non-recursive binary search *bsearch(x,A)* that takes as input a value x and a sorted array A with length n=A.length and A[1] <= A[2] <= … <= A[n], and returns the largest index i such that A[i] <= x, or zero if there is no such index.

6. (16’) (a) Suppose we have a hash function h(k) = k mod 11 and a hash table T of size 11. Illustrate the insertion of keys 44, 77, 30, 92, 100, 54, 63 into T, using chaining as the collision resolution technique.

(b) Solve part (a) again, but using linear probing as the collision resolution technique.

7. (a) (6') Suppose you're talking to a student who have learnt binary search trees, but has never heard of red-black trees. Give a concise, convincing argument why this student would ever want to learn about red-black trees and how they achieve improved behavior over BSTs. (If you run out of room, you're not being concise enough.)

(b) (8') Write pseudocode to implement the function BH(x), that returns the black height of node x in a red-black tree.

8. (12’) Bucket Sort with a Twist. Bucket sort assumes its inputs are uniformly distributed and we can easily adapt it for any uniform distribution and get pretty good performance. But what if we have a different kind of distribution? Suppose you have an input data set of integers whose keys follow a distribution such that

- All values fall in [0, 100).
- There’s a 64% chance values will fall in [70, 100). Among those values that do, 50% are expected to be in [80, 90) and be uniformly distributed between 80 and 90. All values in [70, 80) and [90, 100) are equally likely to occur.
- Values in [40, 50) are expected 6% of the time and values in this range are uniformly distributed.
- All other values are uniformly distributed.

For the sake of convenience, you may assume that the size of input to be sorted is a multiple of 100. Explain how to adapt bucket sort for this situation so that performance does not suffer. Justify your decisions and note running times