CPSC 331 — Term Test #2

March 26, 2007

Name: ____

Please **DO NOT** write your ID number on this page.

Instructions:

Answer all questions in the space provided.

Point form answers are acceptable if complete enough to be understood.

No Aids Allowed.

There are a total of 50 marks available on this test.

Duration: 90 minutes

ID Number: ____________________________________

Question	Score	Available
1		10
2		10
3		9
4		14
5		7
Total:		50

(10 marks) 1. Short answer questions — you do *not* need to provide any justifications for your answers. Just fill in your answer in the space provided.

(a) True or false: binary search is especially well-suited for searching a dictionary that is implemented as an ordered linked list.

Answer: _____________

(b) True or false: merge sort uses $O(n)$ auxiliary space.

Answer: _____________

(c) True or false: the insertion sort algorithm is faster than merge sort on data which is nearly sorted.

Answer: _____________

(d) True or false: the double hashing technique yields hash functions that are useful with the chaining collision resolution mechanism.

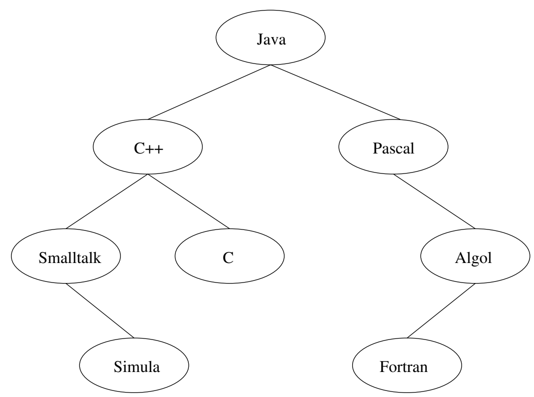
Answer: _____________

(e) Is the level-order traversal of a binary tree best implemented with a stack or a queue?

Answer: _____________

(f) In the binary tree below, which node follows Simula in a preorder traversal? Which node follows Simula in a postorder traversal?

Preorder: ____________ Postorder: ____________



(g) Consider the `search` function for the `dictionary` abstract data type. Using big-Oh notation, fill in the following table to indicate the *worst-case* asymptotic running time as a function of n, where n is the number of entries in the dictionary, assuming that the key being searched for is *not* in the dictionary.

Data Structure	**worst-case running time**
red-black tree	
hash table with chaining	
hash table with open addressing	

2. Consider the red-black tree data structure.

(5 marks) (a) State the five properties satisfied by a binary search tree that is also a red-black tree.

(2 marks) (b) Define the *black height* of a node in a red-black tree.

(3 marks) (c) Give an informal description of how black height is used to prove a worst-case bound on the height of a red-black tree.

3. Consider **hash tables** using the **open addressing with linear probing** collision resolution mechanism, using **table size** $m = 7$ and the hash function

$$h(k, i) = k + ci \pmod{m}, \quad c = 2$$

for which we assume that the key k is an integer.

(2 marks) (a) Draw the hash table (with the above table size and hash function) that would be produced by inserting the following values into an initially empty table:

$$17, 21, 38, 24$$

Your hash table:

0	1	2	3	4	5	6

(3 marks) (b) Describe an algorithm **using pseudocode** that can be used to search for a given value in a hash table with open addressing and linear probing *using the hash function described above.*

(2 marks) (c) Give one example of a set of 4 input values that, when inserted into an empty hash table as described above, would cause a search for the key 11 to result in the worst-case number of operations. Explain why your inputs do in fact result in a worst case for the search for 11.

(2 marks) (d) What is the expected number of comparisons required for an unsuccessful search in the hash table using open addressing with linear probing and $c = 1$? Your answer should be given as a function of both n (number of values stored in the hash table) and m (the size of the hash table). Under what assumption(s) does this estimate hold?

4. The following questions deal with the Heap Sort algorithm.

(2 marks) (a) Draw the binary tree representation of the Max-Heap stored in the following array:

0	1	2	3	4	5
20	10	8	3	5	4

(2 marks) (b) Give an array that represents the Max-Heap obtained after calling **DeleteMax** on the heap from Question 4a.

Your array:

0	1	2	3	4	5

(3 marks) (c) Describe the **MaxHeapify** operation **using pseudocode**.

(3 marks) (d) Give a simple English description (or pseudocode if you prefer) of the **DeleteMax** algorithm.

(4 marks) (e) Give a simple English description (or pseudocode if you prefer) of the **heap sort** algorithm.

5. Consider the following algorithm for sorting an array. Assume that the function inorder(T, B) stores the result of an inorder traversal of a binary search tree T in an array B.

   TreeSort(A, n)

   PRECONDITION: A is a array of $n \geq 1$ integers
   POSTCONDITION: returns array with the elements of A in increasing order

   Initialize T to be an empty binary search tree
   for i **from** 0 **to** $n - 1$ **do**
   insert($T, A[i]$)
   end for
   inorder(T, B)
   return B

(3 marks) (a) What is the worst-case running time of this algorithm as a function of n, the number of integers in the array A? Justify your answer, and give a characterization of the input that results in the worst case.

(2 marks) (b) What is the average-case running time of this algorithm as a function of n, the number of integers in the array A? Justify your answer.

(2 marks) (c) What is the worst-case running time of this algorithm as a function of n, the number of integers in the array A, assuming that a red-black tree is used instead of a binary search tree? Justify your answer.

Extra page for rough work.

Extra page for rough work.