CPSC 331 — Term Test #1

February 12, 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 45 marks available on this test.

Duration: 90 minutes

ID Number: ____

Question	**Score**	**Available**
1		10
2		8
3		6
4		4
5		10
6		7
Total:		45

(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: black-box tests of the functions of an ADT can be determined by someone who has access to the interface but not the implementation.

Answer: __________

(b) True or false: when used together, the proper use of black-box and white-box testing guarantees the correctness of a program.

Answer: __________

(c) True or false: an algorithm that runs in worst-case time $f(n)$ for inputs of size n is faster for all worst-case inputs than an algorithm that runs in time $g(n)$ if $f(n) \in o(g(n))$.

Answer: __________

(d) True or false: if f and g are functions such that $f \in o(g)$, then $g \in \omega(f)$.

Answer: __________

(e) In general, is the non-empty queue `q` in its original state after the statement `q.enqueue(q.dequeue())` is executed? Yes or no?

Answer: __________

(f) What are the maximum and minimum numbers of leaf nodes in a binary tree with 5 nodes?

Maximum: __________ Minimum: __________

(g) Consider the `insert` function for the `dictionary` abstract data type. Using big-Oh notation, fill in the following table to indicate the asymptotic running time as a function of n, where n is the number of entries in the dictionary, assuming that a search has already been performed to determine whether the element to insert is already in the dictionary.

Data Structure	**worst-case running time**
unordered array	
ordered linked list	
binary search tree	

2. Consider the following algorithm that searches an integer array for a specified value.

   PRECONDITION: `k` is a nonnegative integer, `A` is non-null array of integers
   POSTCONDITION: `idx = -1 OR (idx >= 0 AND A[idx] = k)`

```
int idx = -1
int i = 0
int n = A.length
while i < n AND idx < 0 do
  if A[i] == k then
    idx = i
  end if
  i = i + 1
end while
return idx
```

(1 marks) (a) Give a loop invariant for the loop in this algorithm.

(3 marks) (b) What three properties have to be satisfied by this loop invariant?

(2 marks) (c) Give a **loop variant** for the loop in this algorithm. You do not need to justify your answer.

(2 marks) (d) Use your loop invariant to derive a worst-case bound on the number of iterations of the loop. Simply stating the bound without justifying how it is derived from the loop invariant will earn only 1 mark.

3. Assume that f, g, and g are functions mapping the natural numbers to the natural numbers:

$$f, g, h : \mathbb{N} \to \mathbb{N}.$$

(2 marks) (a) Define **big-Omega** by saying what it means when "$f \in \Omega(g)$." A written definition is required (pictures will receive no marks).

(4 marks) (b) Prove that if $f \in \Omega(h)$ and $g \in \Omega(h)$, then $f + g \in \Omega(h)$.

4. Consider the behavior of the following algorithm when it is given a positive integer n as input:

```
int count = 0
for $i$ from 1 to $n$ do
  for $j$ from $n$ downto 1 do
    for $k$ from 1 to $n/2$ do
      count = count + 1
    end for
  end for
end for
```

(2 marks) (a) Give a function $T(n)$ such that the above algorithm uses $\Theta(T(n))$ steps on input n.

(2 marks) (b) **Briefly** explain how you found the function $T(n)$.

5. The following questions deal with the Stack abstract data type.

(4 marks) (a) Define the stack ADT as presented in class.

(4 marks) (b) Give Java code that implements the push and pop operations efficiently when a singly linked list is used to represent a stack. You may assume the existence of the following private internal class:

```
private class StackNode {
  private Object data;
  private StackNode next;

  private StackNode(Object x, StackNode n)
    { data = x; next = n; }
}
```

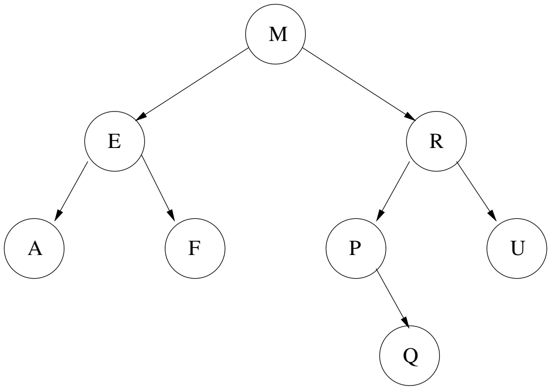
You may also assume that the top of the stack is of type StackNode.

How to Implement the "push" Operation:

How to Implement the "pop" Operation:

(2 marks) (c) Are there any advantages to using a **doubly linked list** to implement a stack? Why or why not?

6. Consider the following binary search tree T:



(3 marks) (a) Draw the binary search tree that would be obtained by

- deleting the node with key M, and
- inserting a node with key S

using the algorithms for insertion and deletion presented in class.

Note: although there are several different binary search trees that could possibly be produced by deleting M and inserting S, to get full credit for this question you must draw the **unique** search tree obtained using the specified algorithms.

(4 marks) (b) Give pseudocode for a **recursive** algorithm that computes the height of a binary tree T. Iterative algorithms will receive at most half credit for this question.

Extra page for rough work.

Extra page for rough work.