NAME: ___________________________________ NETID: _________________________

CS2110 Spring 2015 Prelim 2
April 21, 2013 at 5:30

	0	1	2	3	4	Total
Score	/1	/20	/41	/15	/23	
Grader						

There are 5 questions numbered 0..4 on 8 pages. Check now that you have all the pages. Write your answers in the boxes provided. Use the back of the pages for workspace or for answers if you need more room. Ambiguous answers will be considered incorrect. The exam is closed book and closed notes. Do not begin until instructed. You have 90 minutes. Good luck!

Question 0. (1 points) Write your name and Cornell netid, legibly, on every page.

1. True-false questions (20 points)

a	T	F	Dijkstra's algorithm computes the "all pairs" shortest paths in a directed graph with positive edge weights. That is, running the algorithm a single time, starting from some single vertex, it will compute the min distance from x to y for all nodes x and y in the graph.
b	T	F	An occurrence of keyword **this** in a static method will cause a compile-time error.
c	T	F	A non-abstract subclass of an abstract class C must provide definitions for all the abstract methods of C.
d	T	F	Every directed acyclic graph (dag) has a unique topological sort.
e	T	F	A binary search tree to implement a set of ints is faster than a hash tables in practice.
f	T	F	The invariant of a loop has to be true until the loop condition is false —then the invariant can be false.
g	T	F	If the same element is inserted several times into a Java **List**, the list will contain only a single instance of that element.
h	T	F	Insertion sort of an array of size n has worst-case running time O(n log n)
i	T	F	Breadth-first search uses a stack.
j	T	F	If X is an object of type A, and B is neither a supertype of A nor a subtype of A, then (B)X will cause X to be converted into an object of type B. This is a form of "autoboxing".
k	T	F	If the edges of a connected undirected graph are all different, Prim's and Kruskal's algorithm construct the same spanning tree.
l	T	F	If a graph is drawn so that edges cross, the graph is not planar.
m	T	F	The complexity of inserting an item into a heap containing N items is O(log(N))
n	T	F	Quicksort used to sort a random vector with no repeats has expected performance O(n log n), but worst-case performance O(n*n).
o	T	F	When the weights on all edges of a graph are 1, Dijkstra's shortest-path algorithm does the same thing as DFS (depth-first search).
p	T	F	LinkedList<Integer> is not a subclass of LinkedList<Object>, but Integer[] is a subclass of Object[].
q	T	F	Nested for loops always result in complexity $O(N^2)$
r	T	F	Suppose that X and Y are different objects of type A and that X.equals(Y) is false. Then X.hashcode() != Y.hashcode(). That is, different objects have different hash codes.
s	T	F	Fields have initial values, but local variables do not.
t	T	F	The default layout manager for a JPanel is BorderLayout.

2. Short-answer questions (28 points).

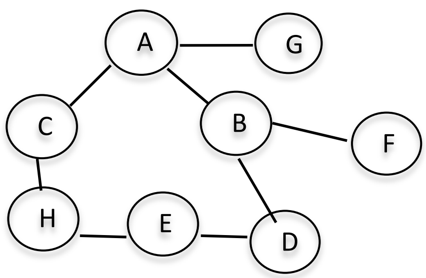
(a) (5 points) Write the definition of asymptotic complexity; that is write down when $f(n)$ is $O(g(n))$ for functions $f(n)$ and $g(n)$.

(b) (5 points) Prove that $n + 100$ is $O(n)$

(c) (5 points) The preorder of a tree with 5 nodes is A B C E D. The inorder is B A E C D. Draw the tree:

(d) (5 points) Consider hashing using open addressing (not chaining, which uses linked lists). When removing an value, the array element it occupies is not set to null; instead, a flag is set to indicate that the element is no long in the set. Explain why in one or two sentences (what would go wrong if we set the array element to null?)

(e) (5 points) List the sequence of nodes visited by BFS (the breadth-first search algorithm) starting at A of the following graph. When a choice of nodes has to be made, choose them in alphabetical order



Put the nodes in the order they are visited here:

(f) (5 points) For the graph in part (e), state the order of nodes visited by DFS (the recursive depth-first search algorithm, not the iterative one), starting at A. When a choice of nodes has to be made, choose them in alphabetical order.

Put the nodes in the order they are visited here:

(g) (5 points) Selectionsort, to sort array b[0..b.length-1], has this invariant:

b[0..k-1] is sorted   and   every value of b[0..k-1] <= every value of b[k..]

Write down the body of the loop. We expect it to be at the same level of abstraction as we have described it in lecture. We do *not* want to see an inner loop.

(h) (6 points) An array segment b[h..k] contains weights on edges of a graph. The weights can be negative, 0, or positive. We want a loop with initialization that swaps the values so that all the negative weights are to the left and all the positive weights to the right. Below is the pre- and post-condition for the loop. Below the pre- and post-condition, draw 3 of the 4 possible loop invariants that arise by combining the pre- and post-condition. *You do not have to put variables in to mark boundaries. Do not write the initialization or the loop*.

	h		k
pre: b		?	

	h		k
post: b	< 0	0	> 0

3. (15 points) Create a min-heap of ints (of maximum size eight) by inserting five values 60, 7, 10, 20, -5 in the order shown (leaving room for three more). Draw the resulting heap first as a tree (exactly as seen in class) and then as it is stored in an array (or ArrayList).

(a) 5 points. The heap:

(b) 5 points. The corresponding array (or ArrayList) (put a slash / through “empty” elements):

(c) 5 points. Suppose poll() is now called once. Put the polled value here ____________ and below, show the array (or ArrayList) after that call on poll().

4. (23 points) **Recursion and asymptotic complexity**. For this question, consider trees whose nodes, of class Node, contain three fields:

value: the value at this node. Its type is some class —not a primitive type
left: left subtree (null if empty left subtree)
right: right subtree (null if empty right subtree)

(a) (6 points) Consider a binary search tree (BST) implemented using class Node. Unfortunately, the person who created a BST used > instead of < for comparison, so the tree got put in kind of backward. For example, the inorder traversal of one BST should have been (1, 3, 4, 5, 6, 9) but was (9, 6, 5, 4, 3, 1). Write method reverse, below, which can be used to rectify the situation. You can use the high-level statement "Swap b and c" to swap the values of two variables b and c. *Hint: values in the nodes should not be changed or compared with each other*.

```
/** Precondition: t is a binary search tree.
  * If t != null, change as many nodes of t as necessary to reverse
  * the inorder traversal of the tree.  */
public static void reverse(Node t) {

}
```

(b) (6 points) Write the body of function equals, given below.

```
/** Return true iff tree t equals tree s –meaning they have the same shape and same values
    in their corresponding nodes. */
public static boolean equals(Node s, Node t) {

}
```

(c) (5 points) Suppose we use this function equals to test equality of M balanced BSTs each containing N nodes. That is, we want to test whether *all* these M balanced BSTs are identical. What is the worst-case complexity of the resulting program, expressed in terms of N and M? Justify your answer by briefly explaining how you arrived at it.

(d) (6 points). Below is a version of depth-first search, explained at a high level of abstraction. The specification is correct —it is what we want— but the algorithm has errors. Fix them. You may explain what is wrong below the algorithm and cross off things in it. Just keep things legible.

```
/** Node u is unvisited. Visit all nodes that are REACHABLE from u. */
public static void dfs(int u) {
   for all edges (u, v) leaving u:
       if (v is unvisited)  {
              visit node v;
              dfs(v);
      }
     visit node u
}
```