CS211 Spring 2005
Final Exam
May 17, 2005

Name——————————————————— **NetId**———————————

Instructions

Write your name and Cornell netid above. There are 15 questions on 12 numbered pages. Check now that you have all the pages. Write your answers in the boxes provided. Use the back of the pages for workspace. Ambiguous answers will be considered incorrect. The exam is closed book and closed notes. Do not begin until instructed. You have $2\frac{1}{2}$ hours. **Good luck!**

	1	2	3	4	5	6	7	8
Score	/5	/4	/4	/7	/20	/4	/4	/10
Grader								

	9	10	11	12	13	14	15	Σ
Score	/6	/6	/5	/5	/10	/6	/4	/100
Grader								

1. (5 points) Consider a sorted circular doubly-linked list where the head element points to the smallest element in the list.

   (a) What is the asymptoptic complexity of finding the smallest element in the list?

   (b) What is the asymptoptic complexity of finding the largest element in the list?

   (c) What is the asymptoptic complexity of determining whether a given element e appears in the list?

   (d) What is the asymptotic complexity of finding the median element in the list?

   (e) What is the asymptotic complexity of deleting a given element e in the list (not including the cost of finding it)?

2. (4 points) Check the ADT or data structure that is most appropriate for each of the following problems:

(a) You want to build an address book with entries in alphabetical order by last name.

☐ Unsorted List
☐ Sorted List
☐ Stack
☐ Balanced Search Tree
☐ Directed Graph

(b) You want to build a meeting reminder for a PDA that keeps track of events you schedule and periodically checks the next event to sound an alarm to remind you of the next thing you need to do.

☐ List
☐ Hashtable
☐ Stack
☐ Queue
☐ Priority Queue

(c) You want to build a table of contents for a textbook. The textbook consists of chapters, chapters consist of sections, and sections consist of subsections.

☐ List
☐ Hashtable
☐ Tree
☐ Binary Tree
☐ Balanced Tree

(d) You want to build an email address miner that scans a hard drive looking for email addresses, then sends them to a remote host.

☐ List
☐ Hashtable
☐ HashSet
☐ Array
☐ Balanced Tree

3. (4 points) Suppose we wish to compute x^n for integer values of n. We can do this recursively as follows:

- if $n = 0$, return 1
- if $n = 1$, return x
- if $n > 1$ and n is even, recursively compute $y = x^{n/2}$ and return $y \cdot y$
- if $n > 1$ and n is odd, recursively compute $y = x^{n-1}$ and return $x \cdot y$.

Assuming addition and multiplication are $O(1)$ operations, what is the worst-case asymptotic complexity of computing x^n this way?

4. (7 points) Prove that for odd $n \geq 1$,

$$\sum_{i=1}^{n} (-1)^{i+1} i^2 \;\; = \;\; 1^2 - 2^2 + 3^2 - 4^2 + 5^2 - \cdots + n^2 \;\; = \;\; \frac{n(n+1)}{2}.$$

5. (20 points) True or false?

   (a) T F The AVL invariant states that a tree's shortest and longest paths differ in length by at most 1.

   (b) T F The rotate operation on AVL trees preserves inorder numbering.

   (c) T F Abstract classes can be instantiated like any other class.

   (d) T F A stack is a FIFO structure and a queue is a LIFO structure.

   (e) T F If class A implements interface I, class B extends A, and class C extends B, then C implements I.

   (f) T F If class A implements interface I, class B extends A, class C extends B, and interface J extends interface I, then C implements J.

   (g) T F 3^n is $O(2^n)$.

   (h) T F $\log(3^n)$ is $O(\log(2^n))$.

   (i) T F $\log_3 n$ is $O(\log_2 n)$.

   (j) T F $\log(n^2)$ is $O(\log(n^3))$.

   (k) T F The optimal number of collisions per bucket in a hashtable is 1.

   (l) T F If $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$, then $f(n)$ is $O(h(n))$.

   (m) T F An occurrence of 'this' in a static method will cause a compile-time error.

   (n) T F Binary search is as efficient on linked lists as on arrays, provided the list is doubly linked.

   (o) T F The finally block in a try-catch-finally statement is executed even if no exception is thrown and there is a return statement in the try block.

   (p) T F A non-abstract subclass of an abstract class C must provide definitions for all the abstract methods of C.

   (q) T F Downcasting is always legal and can be done without an explicit cast.

   (r) T F Every dag has a unique topological sort.

   (s) T F In the worst case, search in an unbalanced binary tree is asymptotically the same complexity as search in a balanced binary tree.

   (t) T F An ActionListener registered with a JMenuItem is executed automatically whenever that JMenuItem is selected.

6. (4 points) Recall that in a binary tree, a node may have 0, 1, or 2 children. In the following questions about binary trees, the *height* of a tree is the length (number of edges) of the longest path. A tree consisting of just one node has height 0.

   (a) What is the maximum number of nodes in a binary tree of height d? ☐

   (b) What is the minimum number of nodes in a binary tree of height d? ☐

   (c) What is the maximum height of a binary tree containing n nodes? ☐

   (d) What is the minimum height of a binary tree containing n nodes? ☐

7. (4 points) Using a witness pair, prove that $\log(n + 42)$ is $O(\log n)$. (*Hint.* Try 2 and 7.)

8. (10 points) Write a recursive `delete` method for singly-linked lists with integer data that deletes the first occurrence of a given integer from the list and returns the resulting list.

```
class ListNode {
    private int value; //data value
    public ListNode next; //next element of list, or null if last
    public ListNode(int v) { value = v; }
    public int value() { return value; }
}
```
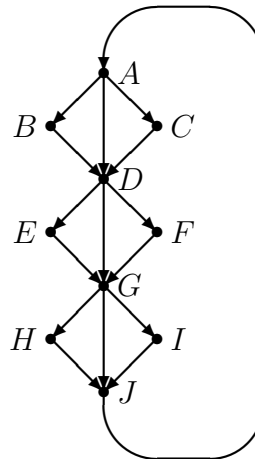
```
//delete first occurrence of i in list s
static ListNode delete(int i, ListNode s) {



}
```

9. (6 points) List the sequence of nodes visited by preorder, inorder, and postorder traversals of the following tree:
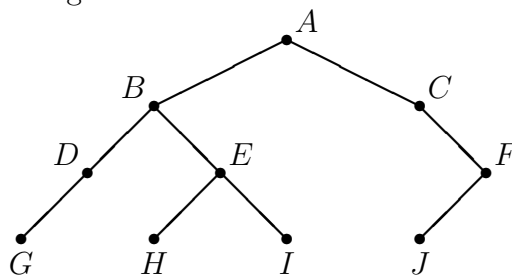
(a) preorder:

(b) inorder:

(c) postorder:

10. (6 points) Consider the following directed graph.

(a) List the nodes in the order they would be visited in a depth-first search of the graph starting at A. When choosing a node to explore next, break ties in favor of the alphabetically least.

(b) List the nodes in the order they would be visited in a breadth-first search of the graph starting at A. When choosing a node to explore next, break ties in favor of the alphabetically least.

(c) How many directed simple cycles does this graph have?

☐ 0 ☐ 1 ☐ 3 ☐ 9 ☐ 27 ☐ infinitely many

11. (5 points) Given the following heap, draw the heap that would result after deleting the minimum element.
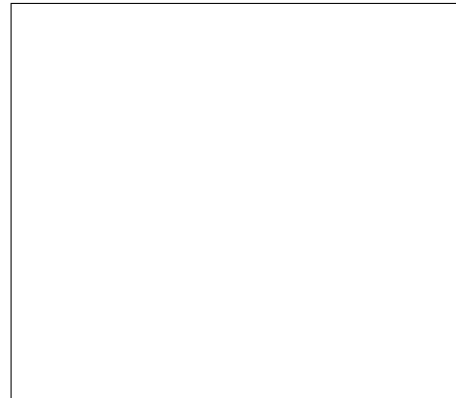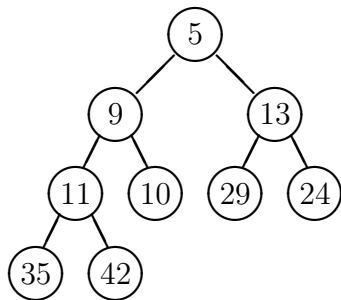
12. (5 points) Say the following tree was obtained by inserting the element 42 into an AVL tree. The tree no longer satisfies the AVL invariant, but the invariant can be reestablished by performing two rotate operations. Show the resulting tree after this is done.
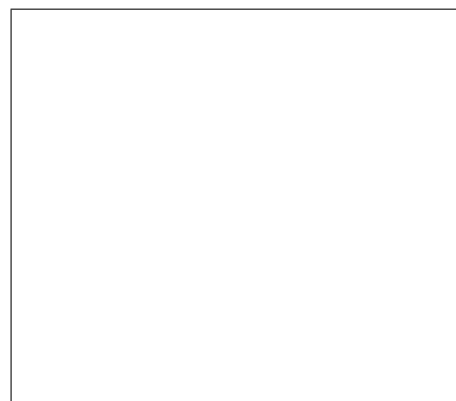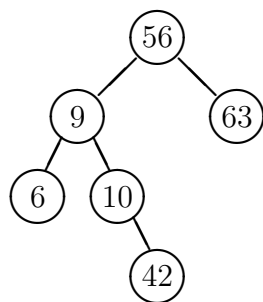
13. (10 points) Given a binary search tree, write an instance method `between(...)` that takes a lower bound `lower` and an upper bound `upper` and returns a `HashSet` containing all of the data elements `x` in the tree between `lower` and `upper`, inclusive. To add an element `x` to the `HashSet` `h`, use `h.add(x)`. (*Hint.* Define a helper method `betweenHelper` that takes an extra `HashSet` argument containing all the elements accumulated so far.)

```java
import java.util.HashSet;

class TreeNode<E extends Comparable> {
    E value;
    TreeNode<E> left;
    TreeNode<E> right;
```

```java
    HashSet<E> between(Comparable lower, Comparable upper) {



    }




}
```

14. (6 points) Say what is output by the following programs. If the program gives an error, say what that error is.

(a)
```
int x = 5;
try {
    x = 3;
    throw new NullPointerException();
} catch (Exception e) {
    System.out.println(x);
} finally {
    System.out.println(x + 1);
}
```

(b)
```
class A { int foo() { return 1; }}
class B extends A { int foo() { return 2; }}
class C extends B { int bar(A a) { return a.foo(); }}
C x = new C();
System.out.println(x.bar(x));
```

(c)
```
class A { int e = 1; }
class B extends A { int e = 2; }
class C extends B { int bar(A a) { return a.e; }}
C x = new C();
System.out.println(x.bar(x));
```

15. (4 points) Questions (a) and (b) on the next page refer to the following constructor for a graphical user interface.

```java
GUI() {
    final JFrame frame = new JFrame("Rock Through the Ages");
    final JMenuBar menuBar = new JMenuBar();
    final JTextArea textArea = new JTextArea();

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JButton button = new JButton("1965-1969");
    button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            frame.setTitle("Rock Through the Ages: 1965-1969");
            textArea.setText("1967 Jimi Hendrix - Axis: Bold As Love\n");
            textArea.append("1967 Cream - Disraeli Gears\n");
            textArea.append("1968 Rolling Stones - Beggar's Banquet");
            frame.validate();
        }
    });
    menuBar.add(button);

    button = new JButton("1970-1974");
    button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            frame.setTitle("Rock Through the Ages: 1970-1974");
            textArea.setText("1972 Yes - Fragile\n");
            textArea.append("1973 Led Zeppelin - Houses of the Holy\n");
            textArea.append("1973 Pink Floyd - Dark Side of the Moon");
            frame.validate();
        }
    });
    menuBar.add(button);
    frame.setJMenuBar(menuBar);

    textArea.setPreferredSize(new Dimension(400,200));
    frame.add(textArea,BorderLayout.CENTER);

    frame.pack();
    frame.setLocation(200,200);
    frame.setVisible(true);
}
```
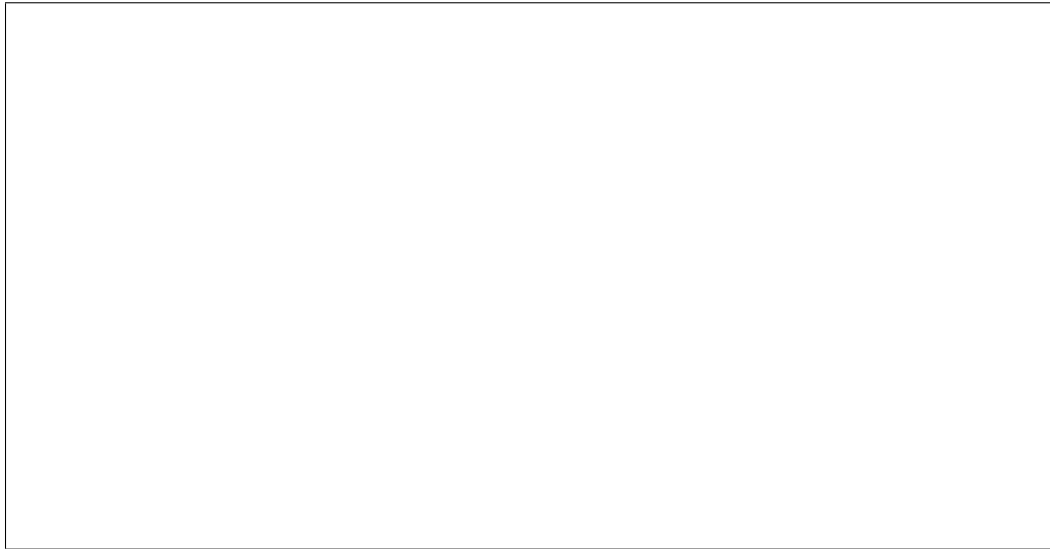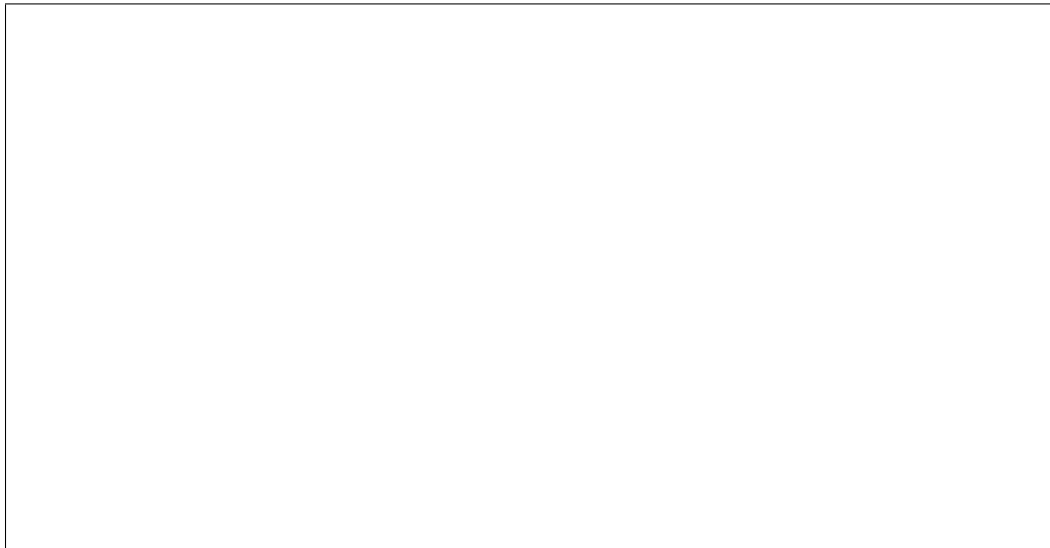
(a) Draw the object that appears on the screen when the code on the previous page is executed.

(b) Draw the object as it appears after the "1970-1974" button is pushed.

End of Exam