

**CMSC132
Fall 2005
Midterm #2**

First Name: _____

Last Name: _____

Student ID: _____

Section time _____ **TA:** _____

Grader Use Only:

#1		(10)
#2		(10)
#3		(15)
#4		(15)
#5		(15)
#6		(15)
#7		(10)
#8		(10)
Total		(100)
Honors		(10)

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: _____

General Rules (Read):

- This exam is closed book and closed notes.
- If you have a question, please raise your hand.
- Total point value is 100 points.
- Answer True/False questions by circling the T or F at the end of the question.
- Answer multiple-choice questions by circling the letter (e.g., a, b) at the front of each choice.
- Answer essay questions concisely using 1 or 2 sentences. Longer answers are not necessary and are discouraged.
- **WRITE NEATLY.** If we cannot understand your answer, we will not grade it (i.e., 0 credit).

Problem 1 Networking (20 pts)

- A. (4 pts) Network architecture
- What is a protocol?
 - Why are protocols useful for a network?
 - Why is the OSI network model designed as 7 separate layers?
 - List 3 examples of programs used at different layers of the internet.
 - What is the difference between an IP address and a URL?
- B. (4 pts) Reliability
- What is the difference between reliable and unreliable network connections?
 - Which of the following layers of the network model may be unreliable?
 - Physical
 - Data-link
 - Transport
 - Network
 - Why do we not simply make all parts of the network reliable?
- C. (4 pts) Packets vs. Connections
- Packets are more reliable than connections T or F
 - Packets are more expensive than connections T or F
 - Packets are simpler than connections T or F
 - Which protocol may be viewed as a connection? UDP or TCP
- D. (4 pts) Client-server model
- Servers can only serve one client T or F
 - A client contacts a server first T or F
 - How do clients distinguish between multiple servers at the same IP address?
 - List examples each 3 server and 3 client programs.
- E. (4 pts) Java support for networking
- What are sockets?
 - What is the main difference between the Socket and ServerSocket classes?
 - What is the main difference between the Socket and DatagramSocket classes?
 - Why is there no DatagramServerSocket class?
 - In Java, why does data from a network appear similar to data from a file?
 - What are applets?

Problem 2 Files, Scanner, and Generics in Java

F. (4 pts) Files & Scanner

- Files may be viewed as streams when opened using either the `FileReader` and `FileInputStream` classes. What is the difference between the two types of streams?
- What is the motivation for using the `Scanner` class to process text input?
- Write code to show how the scanner may be used to read in a series of alternating numbers (ints) and names (Strings) from a text file (starting with a number).

```
class Scanner {  
    Scanner ( ... ) { ... }  
    boolean hasNext() { ... }  
    String next() { ... }  
    int nextInt() { ... }  
    String nextLine() { ... }  
}
```

```
myFileReader() {  
    try {  
        BufferedReader f = new BufferedReader( new FileReader( filename ));  
  
        // insert code to read in numbers and names from file  
  
    } catch (IOException e) {  
        System.out.println(e.getMessage());  
    }  
}
```

G. (4 pts) Generics

- What is the motivation for using generic types?
- Classes in the Java Collection Framework support generic types in Java 1.5. Show how to modify the following code to declare a `LinkedList` class for Strings using generics, then add & get a String object from the list.

```
// previously  
LinkedList myList = new LinkedList();  
myList.add( new String("test") );  
String s = (String) myList.get(0);
```

Problem 3 Algorithmic Complexity (20 pts)

- H. (4 pts) Algorithmic complexity
- a. What is algorithmic complexity?
 - b. List 2 reasons benchmarking is better than analyzing complexity
 - c. What is the difference between best case, worst case, and average case?
 - d. What is a recurrence relation?
- I. (4 pts) Big-O notation and complexity
- a. What does the Big-O notation represent?
 - b. Why are $O(2n+4)$ and $O(n)$ considered equivalent?
 - c. Why are $O(n^2)$ and $O(n)$ not considered equivalent?
 - d. What are some simple rules for calculating Big-O notation?
 - e. Sort the following complexity categories from least to most complex
 - i. Constant $O(1)$
 - ii. Cubic $O(n^3)$
 - iii. Exponential $O(2^n)$
 - iv. Linear $O(n)$
 - v. Logarithmic $O(\log(n))$
 - vi. Quadratic $O(n^2)$
 - f. How are the following complexity categories similar?
 $O(n^2), O(n^4), O(n^3), O(n^{12}), O(n^{99})$
- J. (4 pts) Calculating Big-O functions

What is the asymptotic complexity of the function $f()$ below (using big-O notation) when the complexity of $f()$, and $g()$ are as shown?

$f(n) \{$
 $g(n);$
 $h(n);$
 $\}$

- a. $h(n) = n+1, g(n) = 2n$ $f(n) = O(\quad)$
- b. $h(n) = 3n, g(n) = 4n^2$ $f(n) = O(\quad)$
- c. $h(n) = \log(n), g(n) = 5\sqrt{n}$ $f(n) = O(\quad)$
- d. $h(n) = 8\sqrt{n}, g(n) = 2n$ $f(n) = O(\quad)$

Problem 4 Recursive Algorithms (20 pts)

L. (4 pts) Recursion

- Describe the difference between an iterative and recursive algorithm?
- What are the 2 main parts of a recursive algorithm?
- Name 4 requirements for recursive algorithms to work correctly.
- Name 2 advantages & 2 disadvantages of recursive algorithms

M. (4 pts) Legality of recursive code

For each of the following codes, describe what result is returned when foo(n) is invoked. If no result is returned, explain why.

a.

```
int foo (int n) {
    return foo(n-1);
}
```

 foo(n) =

b.

```
int foo (int n) {
    if (n == 1) return 1;
    return foo(n-1);
}
```

 foo(n) =

c.

```
int foo (int n) {
    if (n == 1) return 1;
    return 1+foo(n-1);
}
```

 foo(n) =

d.

```
int foo (int n) {
    if (n == 1) return 1;
    return foo(n);
}
```

 foo(n) =

e.

```
int foo (int n) {
    if (n == 1) return 1;
    return foo(n-1)+f(n-1);
}
```

 foo(n) =

N. (4 pts) Writing recursive code (you may use helper functions)

- Write a recursive function to search an unsorted array for a number k
`public static boolean findNum(int k, int[] array)`
- Write a recursive function to calculate the sum of an array of ints
`public static int sumArray(int[] array)`
- Write a recursive function to determine whether an array of ints is sorted (ascending)
`public static boolean sortedArray(int[] array)`

Problem 5 Linear Data Structures (20 pts)

- O. (4 pts) Taxonomy & properties
- Describe the main difference between linear and hierarchical data structures
 - Describe the main difference between a linked list and an array
 - Describe the main difference between a queue and a stack
 - Describe a circular linked list
- P. (4 pts) Given the following Java class definition for a singly linked list

```
Class Node {
    int myValue;
    Node next;
}

Class LinkList {
    Node head; // first node in list
    Node find(int k) { ... }
    void insertHead(Node n) { ... }
    void insertTail(Node n) { ... }
    void removeTail() { ... }
}
```

Write the following code:

- `find(k)` – Find a node with `myValue = k` in a linked list
- `insertHead(n)` – Insert node `n` at the head of a linked list
- `insertTail(n)` – Insert a node `n` at the tail of a linked list
- `removeTail()` – Delete the node at the tail of a linked list
- Which 2 methods in the `LinkList` class can be used to implement a queue?
- Which 2 methods in the `LinkList` class can be used to implement a stack?

Problem 6 Trees (20 pts)

Q. (4 pts) Binary search trees (BST)

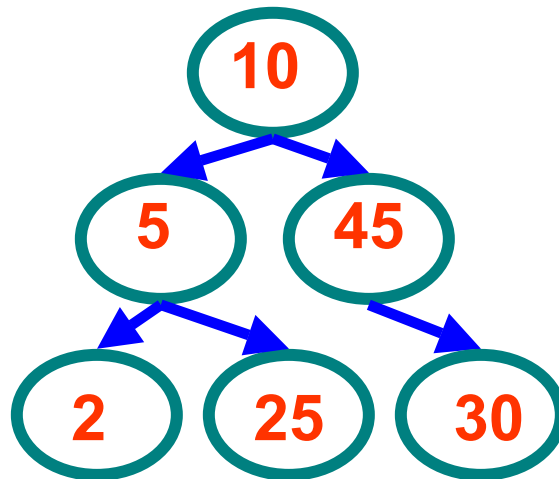
- a. What is the key property of a binary search tree?
- b. On average, what is the complexity of doing an insertion in a binary search tree?
- c. On average, what is the complexity of doing a find in a binary search tree?
- d. What is the worst-case complexity of doing a find in a binary search tree?
- e. What can cause worst-case behavior in a binary search tree?

R. (4 pts) Binary search trees examples

- a. Draw the binary search tree created when the following values are inserted in order:
6, 5, 2, 8, 10, 3
- b. Given the previous BST, draw the tree created when the node 5 is removed

S. (4 pts) Traversals

- a. What is a tree traversal?
- b. What is the difference between a depth-first and breadth-first traversal?
- c. Pre-order, in-order, and post-order are all depth-first traversals T or F
- d. Pre-order traversals are faster than post-order traversals T or F
- e. For the following binary tree, provide an example of a
 - i. Preorder traversal
 - ii. Postorder traversal
 - iii. Breadth-first traversal



T. (4 pts) Given the following Java class definition for a binary tree

```
Class Node {
    int myValue;
    Node left;
    Node right;
}

Class Tree {
    Node root;           // root node in tree
    Node find(int k) { ... }
    int treeHeight() { ... }
    void preorder() { ... }
}
```

Write the following code:

- a. `find(k)` – Use a recursive algorithm to find a node with `myValue = k` in tree
- b. `treeHeight()` – Use a recursive algorithm to find the height of a tree
- c. `preorderPrint()` – Use a recursive algorithm to print `myValue` for each node using a preorder traversal of the tree, starting from the root

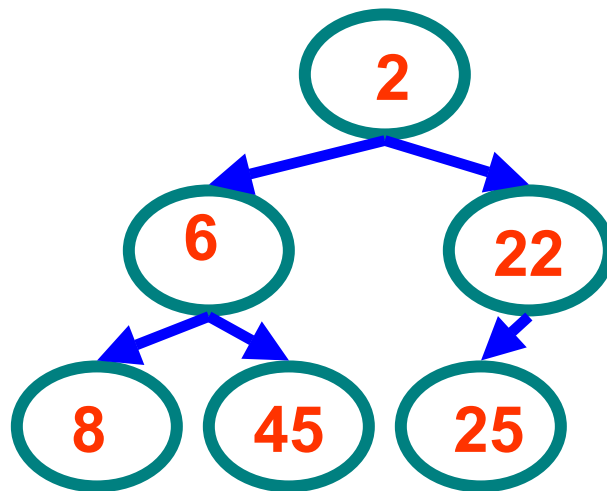
Problem 7 Heaps (20 pts)

U. (4 pts) Properties & characteristics

- a. What are two key properties of a heap?
- b. What operation(s) supported by binary search trees are not supported by heaps?
- c. On average, what is the complexity of doing an insertion in a heap?
- d. On average, what is the complexity of doing a find in a heap?
- e. What is the worst-case complexity of doing a find in a heap?
- f. How can heaps be stored in a compact array representation?
- g. Why are heaps used to implement priority queues?

V. (4 pts) Examples

- a. Draw the heap created when the following values are inserted in order:
6, 5, 2, 8, 10, 3
- b. Given the previous heap, draw the heap produced when the smallest value is removed
- c. Show the tree representation of the following heap (in array representation)
A, B, C, D, E, F
- d. Show the array representation of the following heap (in tree representation)



Problem 8 Maps & Hashing (20 pts)

W. (4 pts) Sets & maps

- a. What is the difference between a set and a map?
- b. What is the difference between a map and a hash table?
- c. How are maps useful?
- d. Given the following TreeMap API, show how to write code to construct a TreeMap storing String objects for actor names with movie titles. The TreeMap should allow movie names to be used to look up lead actor names (assuming each movie has a single leading actor)

```
public class TreeMap {
    TreeMap() { ... }
    Object get( Object key ) { ... }
    Object put( Object key, Object value ) { ... }
    Object remove( Object key ) { ... }
}

TreeMap myDB = new TreeMap();

void addMovie( String leadActor, String movie ) {
    ... // write code to add leading actor name for movie to TreeMap
}

String findLeadActor( String movie ) {
    ... // write code to find String for leading actor name for movie
}
```

X. (4 pts) Hashing

- a. What is a hash function?
- b. What is a desirable property of hash functions?
- c. What is a perfect hash function?
- d. What is a collision?
- e. What is the difference between open addressing and chaining (bucket hashing)?
- f. What happens when an open addressing hash table is close to full?
- g. What is the relationship between the hashCode() and equals() methods in Java?