CS 307 – Midterm 2 – Spring 2002

Name____________________________________

Last 4 digits of SSN / Student ID ______________

Class Unique ID ___________________________

Instructions:
1. There are 6 questions on this test.
2. You will have 2 hours to complete the test.
3. You may not use a calculator on the test.
4. When code is required, write Java code.
5. The style guide is not in effect.
6. Ensure your answers are legible.
7. When writing code you may not use any methods or classes from the Java Standard Library except as noted and the System.out.print, System.out.println, and the equals method.

1. (2 points each, 20 points total) Java Mechanics and theory. If an error would occur answer "syntax error" or "runtime error" depending on what type of error it would be.

A. What is the average case Big O for inserting N items into an initially empty Linked List that is to be maintained in sorted order?

_____________________________________________________________________________

B. What is the Big O for deleting a node from the end of a singly linked list that has a head and tail reference?

_____________________________________________________________________________
C. Consider the following code from class M2:

```java
public static int stone(int n) {
    if(n <= 0)
        return 3;
    else
        return n + stone(n - 2);
}
```

What is printed out by the following code?

```java
System.out.println( M2.stone(8) );
```

D. Consider the following code from class M2:

```java
public static int mav(int m) {
    if(m <= 0)
        return 2;
    else
        return 2 + mav(m - 1) + mav(m - 2);
}
```

What is printed out by the following code?

```java
System.out.println( M2.mav(5) );
```

E. Given a positive integer N, what is the Big O of M2.mav( N ) from question 1.D?
F. Consider a queue of ints that uses a native array of ints, myContainer, to store the elements of the queue. myContainer starts with a capacity of 5. It is resized only when necessary. The queue uses wraparound as discussed in class. Draw myContainer and show its contents after the following code is executed. Be sure to clearly label the front and back elements of the queue.

```java
Queue q = new Queue();
q.enqueue(12);
q.enqueue(9);
q.dequeue();
q.enqueue(73);
q.enqueue(81);
q.dequeue();
q.enqueue(55);
q.dequeue();
q.enqueue(10);
```

G. Consider the following code:

```java
for( int i = 0; i < N; i++ )
    for( int j = 1; j < N; j *= 2 )
        west( i );
```

method west has a Big O of O( N ).

What is the Big O of the above code?
H. Consider the following code from class M2:

```java
class M2 { 
    public static int warrior(int n) { 
        if(n == 0) 
            return 3; 
        else 
            return n + warrior(n - 2); 
    } 
}
```

What is printed out by the following code?

```java
System.out.println(M2.warrior(7));
```

---

I. Consider a stack that stores integers. What is output by the following code:

```java
Stack s = new Stack(); 
s.push(12); 
s.push(2); 
s.push(8); 
s.push(31); 
s.pop(); 
s.push(s.top()); 
s.push(s.top()); 
for(int i = 0; i < 3; i++) 
    s.pop(); 
System.out.println(s.top());
```

---

J. Consider a singly Linked List class. If the class is to be used as the storage container for a Queue what attributes must the Linked List class have so that all queue operations (front, enqueue, dequeue, and size) are all O(1) operations?
2. (15 points) Complete a removeBack method for a singly linked list class.

Use the following ListNode class

```java
public class ListNode
{
    // returns next node in list or null if no next node
    public ListNode getNext()
    {
        // set next node to parameter
        public void setNext(ListNode next)
    }
    // other methods, implementation, and private data members
    // not shown
}
```

This LinkedList class has a reference to the head node and tail node. The list nodes have references to the next node, but not the previous node. The last node of a nonempty list has its next reference set to null. An empty list is signified by the head and tail pointers both being set to null. There is also a private integer to track the number of elements in the list.

You may not use any of the other methods in the LinkedList class to complete the removeBack method.

```java
public class LinkedList
{
    private ListNode myHead;
    private ListNode myTail;
    private int iMySize;
    // other methods not shown
    // pre: isEmpty() == false;
    // post: getSize() = old getSize - 1, last node removed.
    public void removeBack()
    {
        // complete this method on the next page.
    }
}
```
// pre: isEmpty() == false;
// post: getSize() = old getSize - 1, last node removed.
public void removeBack()
{
3. (20 points) Complete a method to print out all possible subsets of numbers in an array of integers.

Given the array of integers:

```java
int[] iList = { 2, 5, 6 };```

one possible form of the output is:

```
[]
[ 2 ]
[ 5 ]
[ 6 ]
[ 2, 5 ]
[ 2, 6 ]
[ 5, 6 ]
[ 2, 5, 6 ]
```

For this question you may use the ArrayList class. You may only use the following methods from the ArrayList class.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayList()</td>
<td>Constructs an empty list.</td>
</tr>
<tr>
<td>boolean add(Object o)</td>
<td>Appends the specified element to the end of this list.</td>
</tr>
<tr>
<td>Object get(int index)</td>
<td>Returns the element at the specified position in this list.</td>
</tr>
<tr>
<td>boolean isEmpty()</td>
<td>Tests if this list has no elements.</td>
</tr>
<tr>
<td>Object remove(int index)</td>
<td>Removes the element at the specified position in this list.</td>
</tr>
<tr>
<td>Object set(int index, Object element)</td>
<td>Replaces the element at the specified position in this list with the specified element.</td>
</tr>
<tr>
<td>String toString()</td>
<td>Returns a string representation of this list.</td>
</tr>
</tbody>
</table>

You may also use the Integer class, but only the following methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer(int value)</td>
<td>Constructs a newly allocated Integer object that represents the primitive int argument.</td>
</tr>
<tr>
<td>int intValue()</td>
<td>Returns the value of this Integer as an int.</td>
</tr>
</tbody>
</table>

Complete the method on the next page:
// pre: nums != null, no given integer appears more than once in
// nums
// post: all possible subsets of values in nums printed out to
// standard output
public void printSubsets( int[] nums )
// You may implement a helper method if you wish.
4. (15 points) Consider the following code:

```java
public static void rock( int[] nums )
{
    int tmp = 0;
    for( int i = nums.length / 2; i > 0; i = i / 2 )
    {
        for( int k = i; k < nums.length; k++ )
        {
            tmp = nums[k];
            int m = k;
            for( ; m >= i && tmp < nums[ m - i ]; m -= i )
                nums[m] = nums[m - i];
            nums[m] = tmp;
        }
        if( i == 4 || i == 2 )
            printArray( nums );
    }
}

public static void printArray( int[] nums )
{
    // print integers in nums on one line
    for( int k = 0; k < nums.length; k++ )
        System.out.print(nums[k] + " ");
    System.out.print("\n"); // move to next line
}
```

rock and printArray both appear in a class called M2. what is output by the following code?

```java
int[] intList = { 13, 39, 12, 571, 101, 7, 2, 102};
M2.rock( intList );
```
5. (15 points) A Bag is the simplest of the Abstract Data Types. It represents a collection of data. There is no implied order to the data and items may appear more than once. Complete a method that determines the number of items in the Bag equal to a given Object.

```java
public class Bag {
    // storage container for items. Items are
    // in the first iMySize elements.
    private Object[] myItems;

    // number of items in the Bag. May be less than or equal to
    // myItems.length
    private int iMySize;

    // other methods and private data members not shown

    // pre: data != null
    // post: returns number of items in bag equal to data
    public int getCount(Object data) {
        // complete this method below:
    }
}
```
6. (15 points) In a pressurized water nuclear reactor, water is heated by passing through channels between plates of metal that contain the nuclear fuel. The nuclear reactions taking place in the fuel generate large amounts of heat that is transferred to the water. Monitoring temperatures in the reactor is very important as you can imagine, because if the temperature gets too high the reactor may be damaged. A series of temperature instruments measures the temperature of the water at various spots in each cooling channel. Here is cross section of the reactor. The x's indicate possible locations of temperature instruments.

```
x x x x Water x x x x
Fuel
x x x x Water x x x x
Fuel
x x x x Water x x x x
Fuel
x x x x Water x x x x
```

Write a method that analyzes the raw temperature data from the sensors. The temperature data is stored in a 2 dimensional array of ints. Each row stores the readings from one cooling channel. Your method shall return the an average temperature and location of the channel or row with the highest average temperature as well as the temperature and location, row and column, of the sensor with the highest reading. When figuring these values only consider temperatures within the bounds of maxTemp and minTemp inclusive. These are parameters sent to the method.

```java
/* pre: temps != null, temps is a rectangular matrix. (all rows have same number of columns)
 post: return an array of ints. element 0 contains number of row
 with highest average temperature, element 1 contains average temp
 of row with highest average temp (truncated to nearest int),
 element 2 contains the row and element 3 contains the column of
 the highest temp present within the bounds of maxTemp and minTemp,
 and element 4 contains the highest temp within the bounds of
 maxTemp and minTemp.
 */
public int[] getHighTemps(int[][] temps, int maxTemp, int minTemp)
{ // complete the method on the next page.
```
public int[] getHighTemps(int[][] temps, int maxTemp, int minTemp)
{