UNIVERSITY OF WATERLOO

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ECE 250 - ALGORITHMS AND DATA STRUCTURES

Midterm Examination Instructor: R.E.Seviora 4:30-6:00 PM, Oct 26, 1999

Name:..... Student Id:.....					
1.	**2.**	**3.**	**4.**	**5.**	**TOTAL:**

Do **all** problems.The number in brackets[..] denotes the relative weight of the problem (out of 100). If information appears to be missing from a problem, make a reasonable assumption, state it and proceed. If the space for an answer is not sufficient, use the last (overflow) page.
Closed book; no calculators. No "magic" answers will be given credit.

PROBLEM 1 [20]

A. In order to evaluate running times of programs implementing algorithms, we must know something about the computer on which the program will execute. Two models of the computer (in the Java VM context) were considered in this course, the detailed one and the simplified one.

Give the timing parameters of the detailed model. For brevity, give the answers in the manner shown for fetch and store.

1. fetch, store: τ_{fetch}, τ_{store}
2.
3.
4.
5.
6.

B. Determine the running time predicted by the detailed and the simplified computer model presented in the lectures for the following program fragment (where $n \geq 1$):

```
1       for (int i=1; i<n; i*=2)
2          ++k;
```

Statement	*Running Time: Detailed Model*	*Running Time: Simplified Model*
TOTAL		

C. To save paper, this problem is stated at the bottom of page 5.

PROBLEM 2 [20]

A. Consider two non-negative functions f(n) and g(n). For the following three cases of lim f(n)/g(n), $n \to \infty$, state whether $f(n) = \Theta(g(n))$ is true or false. Explain your answer for the last case.

$$f(n) = \Theta(g(n))$$

$\lim f(n)/g(n) = 0$

$\lim f(n)/g(n) = \infty$

$\lim f(n)/g(n) = K,\ 0 < K < \infty$.

B. Consider the Java program fragments given in (i) and (ii). Assume that n,m, and k are nonnegative ints and that the worst-case running time of method `f(n,m,k)` is O(n+m). Determine a tight, big Oh expression for the worst-case running time of the following program fragments:

```
1  for (int i=0; i<n; ++i)
2     for (int j=1; j<n; ++j)
3        f(n,m,k);
```

Statement	*Time*
TOTAL	

C. You have just come out of a meeting, in which a colleague showed that an algorithm he devised was O(n). He gave the running time of the algorithm as T(n) = 5n(10 - 5log(n) + n). He then referred to the definition of big-Oh [..f(n)=O(g(n)) ... n_0 and c ... for all $n \geq n_0$ ... $f(n) \leq cg(n)$..]. He let n_0=10, c=10n and ran a couple of numeric checks:

n	f(n)	cg(n)=cn
10	750	1,000
100	10,000	100,000
1000	975,000	10,000,000

He observed that the bound seemed to be loose, but that he did not have the time to find a tight bound.

Was his argument (for T(n) = O(n)) correct? If your anser is no, identify the errror in his reasoning. If your ansswer is yes, give a tight bound.

PROBLEM 3 [20]

An *upper triangular* matrix is an *n x n* matrix in which all the entries below the diagonal are zero:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} \\ & 0 & a_{33} & a_{34} & a_{35} \\ 0 & & 0 & a_{44} & a_{45} \\ 0 & 0 & & 0 & a_{55} \end{bmatrix}$$

To save space, we will store only the entries on and above the diagonal, in a one-dimensional array: $\{a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{22}, a_{23}, a_{24}, a_{25},..., a_{44}, a_{45}, a_{55}\}$

In the following, consider the interface `Matrix`:

```
public interface Matrix {
   double get (int i, int j);
   void put (int i, int j, double d);
   Matrix plus (Matrix m);
   ...
}
```

A. Derive a function *f(i,j)* which maps a pair of indices *i,j (i ≤ j)* into the corresponding array subscript. For example, *f(1,1)* = 0, *f(1,2)* = 1, ...

B. Consider the Java class `UpperDiagonalMatrix`, which implements `Matrix`. Write an implementation for its constructor and for `get` method. This method returns value of the corresponding matrix entry. Your program should reject invalid array subscripts and throw an `IllegalArgumentException`.

```
public class UpperDiagonalMatrix
  implements Matrix {
  protected int size;
  protected double[] array; //stores the values of matrix elements
  public UpperDiagonalMatrix (int size){



  }
  double get (int i, int j) {
```

PROBLEM 4 [20]]

A. The lectures on abstract data types and design pattersn introduced the concepts of Visitor and Enumeration. The corresponding interfaces were:

```
public interface Visitor {
  void visit (Object object);
  boolean isDone();
}
```

```
public interface Enumeration {
   boolean HasMoreElements();
   Object nextElement () throws ..;
}
```

The class `AbstractVisitor` implements the interface `Visitor`.

Summarize, briefly, the main difference(s) between a `Visitor` and an `Enumeration`.

B. In the lectures, we discussed the class DenseMatrix. This class used an array of arrays of doubles to store the elements of an mxn matrix. In this problem, you will consider a more general matrix whose elements are of the type Object, rather than doubles. One can view such matrices as containers.

Write a `getEnumeration` method for this class. (Hint: recall `getEnumeration` for, e.g., the StackAsArray class, and extend it to this two-dimensional case.)

```
public class DenseObjectMatrix {
  protected int noRows;          //number of rows
  protected int noCols;          //number of columns
  protected Object [][] array;   //array of refs to matrix objects
  //other methods
  //...
  public Enumeration getEnumeration() {
         return new Enumeration () {
```

C. Consider a container that we know contains only instances of the `Int` class. (The `Int` was defined in the lectures. It is a wrapper for the primitive type `int`. Its method `intValue()` returns the value of the integer wrapped.)

Write the code for a `MaxIntVisitor` that finds the maximum value of all `Int`'s in the container. Include also the accessor `getMaxInt`, which *returns* the maximum value of all integers in the container. (Naturally, an instance of `MaxIntVisitor` must have 'visited' the container for `getMaxInt` to return the correct value.)

```
public class MaxIntVisitor
  extends AbstractVisitor {
  //fields

  //methods
  public void visit (Object obj) {









  public boolean isDone() {



  public int getMaxInt () {
```

PROBLEM 1 [cont'd from page 1]

C. You are designing an algorithm whose implementation will executed in a single-chip microcomputer in an embedded system. Because of its internal design, this computer will take a relatively long time to multiply and divide two integer numbers, i.e. $\tau_{\times}$, $\tau_{\div}$ is much larger than τ_{+}, $\tau_{<}$,... In your predictions of running time, you must take this into account. Unfortunately, the available literature does not say how long the time to multiply two integers is, and your web search has not yielded anything useful. You will have to measure such time experimentally.

Write a Java program which would measure $\tau_{\times}$ and print the measured value. You may use the `java.lang.System` method `public static long currentTimeMillis()`, which returns the current time in milliseconds.

PROBLEM 5 [20]

A. The array-based implementation of stack introduced in the lectures used a fixed-length array. In this case, it is possible for the stack to become full.

Rewrite the `push` method so that it doubles the length of the array when the stack is full. For your reference, the original code for `push` is shown below:

```
public class StackAsArray
  extends AbstractContainer implements Stack {
  protected Object[] array;
  //other methods ...
  /* old push
  public void push (Object obj) {
        if (count == array.length)
           throw new ContainerFullException();
        array [count++] = obj;
  }
  */
  public void push (Object obj) {}
```

B. The interface Queue discussed in the lectures contained only the basic queue operations `enqueue, dequeue`, and also `getHead`. In some applications, the method `reverse()` is required. This method will reverse the order of items on the queue. For example, if the original queue containted a,b,c,d (in this order), `reverse()` would reorder its contents to d,c,b,a. Show an algorithm for `reverse()`, for the linked list implementation of the queue.

```
public class QueueAsLinkedList
  extends AbstractContainer implements Queue {
  protected LinkedList list;
  //standard methods
  ...
  //reverse(): reverses the order of items on the queue
  public void reverse () {
```

OVERFLOW SHEET [Please identify the question(s) being answered.]