Exam Number:

| Points off | 1 | 2 | 3 | 4 | | Total off | Net Score |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

# CS 307 – Midterm 2 – Fall 2007

Name________________________________________

UTEID login name _______________________________

TA's Name: David Joseph Ola (Circle One)

Instructions:

1. Please turn off your cell phones and other electronic devices.
2. There are 4 questions on this test.
3. You have 2 hours to complete the test.
4. You may not use a calculator on the test.
5. When code is required, write Java code.
6. When <u>writing</u> methods, assume the preconditions of the method are met.
7. In coding question you may add helper methods if you wish.
8. After completing the test please turn it in to one of the test proctors and show them your UTID.

1. (2 points each, 30 points total) Short answer. Place you answers on the attached answer sheet.

- If the code contains a syntax error or other compile error, answer “compile error”.
- If the code would result in a runtime error / exception answer “Runtime error”.
- If the code results in an infinite loop answer “Infinite loop”.

Recall that when asked for Big O your answer should be the most restrictive correct Big O function. For example Selection Sort has an average case Big O of O(N^2), but per the formal definition of Big O it is correct to say Selection Sort also has a Big O of O(N^3) or O(N^4). I want the most restrictive correct Big O function. (Closest without going under.)

A. What is the output of `System.out.println( a(5) );`

```
public int a(int n){
     if( n < 1 )
          return 2;
     else
          return 1 + a(n - 1);
}
```

B. What is the output of `System.out.println( b("steve") );`

```
public String b(String s){
     if( s.length() <= 3 )
          return "end";
     else
          return s.charAt(0) + b( s.substring(1) ) + s.charAt(0);
}
```

C. What is the output of `System.out.println( c(4) );`

```
public int c(int n)
{     if( n <= 0 )
            return 3;
      else
            return n + c(n - 1) + c(n - 1);
}
```

D. What is the Big O of method `d`? N = `values.length`.

```
public int d(int[] values){
      int total = 0;
      for(int i = 0; i < values.length / 2; i++)
            total += values[i];

      return total;
}
```

E. What is the Big O of method `e`?

```
public int e(int N){
      int sum = 0;
      for(int i = 0; i < N; i++){
            for(int j = 0; j < N; j++){
                  sum += (i - 1) * (j - 1);
            }
      }

      for(int i = 0; i < N; i++)
            sum -= i;

      return sum;
}
```

F. What is the worst case Big O of method `f`? N = mat.length

```
//pre: mat is a square matrix
public int f(int[][] mat, int row, int col){
      int total = 0;

      for(int r = row - 2; r <= row + 2; r++)
            for(int c = col - 2; c <= col + 2; c++)
                  if( 0 <= r && r < mat.length && 0 <= c && c < mat.length)
                        total += mat[r][c];

      return total;
}
```

G. What is the worst case Big O of method `g`? `list` is a LinkedList from the Java standard library. N = `list.size()`

```
public int g(LinkedList<Integer> list, int tgt){
      int found = 0;

      for(int i = 0; i < list.size(); i++)
            if( list.get(i)== tgt )
                  found++;

      return found;
}
```

H. What is the Big O of method `h`? Method `compute` is O(1). N = `values.length`.

```
public int h(int[] values){
      int diff = 0;;

      for(int i = 0; i < values.length; i++){
            for(int j = 1; j < values.length; j *= 3){
                  diff += compute(i, j);
            }
      }

      return diff;
}
```

I. What is the Big O of method `c`?

```
public int c(int N)
{     if( N <= 0 )
            return 3;
      else
            return N + c(N - 1) + c(N - 1);
}
```

J. What is output when method `j` is called? Assume the `Stack` class implements a traditional stack.

```
public void j(){
      Stack<Integer> st = new Stack<Integer>();
      for(int i = 5; i > 0; i--)
            st.push( i + 2 );

      for(int i = 0; i < 3; i++){
            System.out.print( st.pop() );
      }
}
```

K. Consider the following methods from the Java ArrayList class.

| **Method Summary** | |
|---|---|
| `void` | `add(E o)`<br>Appends the specified element to the end of this list. |
| `void` | `add(int index, E element)`<br>Inserts the specified element at the specified position in this list. |
| `E` | `get(int index)`<br>Returns the element at the specified position in this list. |
| `E` | `remove(int index)`<br>Removes the element at the specified position in this list. Return the element that is removed from the list. |
| `E` | `set(int index, E element)`<br>Replaces the element at the specified position in this list with the specified element. Returns the element that was previously at index. |
| `int` | `size()`<br>Returns the number of elements in this list. |

What is output when method `k` is called?

```
public void k(){
     ArrayList<Integer> lt = new ArrayList<Integer>();
     for(int i = 0; i < 6; i++)
          lt.add( i );

     for(int i = 0; i < lt.size(); i++)
          lt.remove( i );

     System.out.println( lt.size() == 0 );
}
```

L. What is output when method `el` is called?

```
public void el(){
     ArrayList<Integer> list = new ArrayList<Integer>();
     list.add( 4 );
     list.add( 6 );
     list.add(0, 7);
     list.add(2, 5);
     list.set(1, 8);
     for(int i = 0; i < list.size(); i++)
          System.out.print( list.get(i) );
}
```

M. What is output when method `m` is called? Assume the `Stack` class implements a traditional stack.

```
public void m(){
      Stack<Integer> st = new Stack<Integer>();
      st.push( 3 );
      st.push( 5 );
      st.push( st.top() );
      st.push( st.top() - 4 );

      while( !st.isEmpty() )
            System.out.print( st.pop() );
}
```

N. A method is O(N). It takes 16 seconds for the method to complete on a data set with 200,000 items. What is the expected time for the method to complete with a data set of 50,000 items?

O. Method `selSort` implements the selection sort algorithm. It takes 10 seconds for the method to complete given a data set of 100,000 integers in random order. What is the expected time for the method to complete given a data set of 400,000 items in random order?

2. (Implementing data structures, 25 points). Write an instance method for a `LinkedList` class named `removeLastOccurrence`. The method removes the last occurrence of a value from the `LinkedList`. The method returns true if the list changed as a result of the method call, false otherwise.

This `LinkedList` class uses singly linked nodes. This `LinkedList` class only contains a reference to the first node in the list. It does not have a reference to the last node in the list. The last node in the list's `next` reference is set to `null`. When the list is empty `head` is set to `null`.

Examples:

```
[A, A, B, C, A, B].removeLastOccurrence(A) -> [A, A, B, C, B]
[A, A, B, C, A, B].removeLastOccurrence(B) -> [A, A, B, C, A]
[A, A, B, C, A, B].removeLastOccurrence(C) -> [A, A, B, A, B]
[D, A, B].removeLastOccurrence(D) -> [A, B]
[D, A, B].removeLastOccurrence(C) -> [D, A, B]
[].removeLastOccurrence(A) -> []
```

**You may not use any other methods in the `LinkedList`. You may create your own helper methods if you wish.**

**You solution must be O(1) space.**

```
public class Node
{     public Node(Object item, Node next)

      public Object getData()

      public Node getNext()

      public void setData(Object item)

      public void setNext(Node next)
}
```

```
public class LinkedList
{    private Node head; // points to the first node in the list
     private int size; // size of list
     // when size == 0, head = null

     // pre: obj != null
     // post: remove the last occurrence of obj in this list.
     // return true if this LinkedList changed as a result of this
     // method call, false otherwise.
     public boolean removeLastOccurrence(Object obj){
     // complete this method on the next page
// pre: obj != null
// post: remove the last occurrence of obj in this list.
// return true if this LinkedList changed as a result of this
// method call, false otherwise.
public boolean removeLastOccurrence(Object obj){
     assert obj != null;
```

3. (Implementing Data Structures, 25 Points) Consider an `AbstractSet` class such as the one you implemented on assignment 8. Implement a method that returns the *Cartesian product* of two AbstractSets.

"The Cartesian product of two sets A and B, denoted by A × B is the set of all ordered pairs (a,b) such that a is a member of A and b is a member of B."

Here are is an example:

```
{1, 2} x {1, 2} = { (1, 1), (1, 2), (2, 1), (2, 2) }
```

In the above example A and B are sets of integers. The Cartesian product of these two sets is a set of ordered pairs of integers. `(1, 1)` is a single element in the resulting set. This should be obvious, but note, the order of elements in an ordered pair matters so the ordered pair `(1,2)` does not equal the ordered pair `(2, 1)`.

Here are some more examples:

```
{1, 2} x {} = {}
{2, 1, 3} x {red, blue} = {(2, red), (2, blue), (1, red),
                           (1, blue), (3, red), (3, blue)}
```

For this question you will use an `OrderedPair` class.

```
public class OrderedPair {
     public OrderedPair (Object first, Object second)

     // other methods not shown
}
```

Recall the `AbstractSet` class does not have an internal storage container. The only methods from `AbstractSet` and `ISet` you may use on this question are as follows:

```
public class AbstractSet implements ISet{

     // returns a new, empty set
     public abstract ISet getEmptySet();

     // add an element to this set
     public abstract boolean add(Object obj);

     // return an iterator for this set
     public abstract Iterator iterator();

     // pre: other != null
     // post: neither this set or other is changed.
     // returns a Cartesian Product of this set and other.
     public ISet cartesianProduct(ISet other){
          //to be completed on the next page
}
```

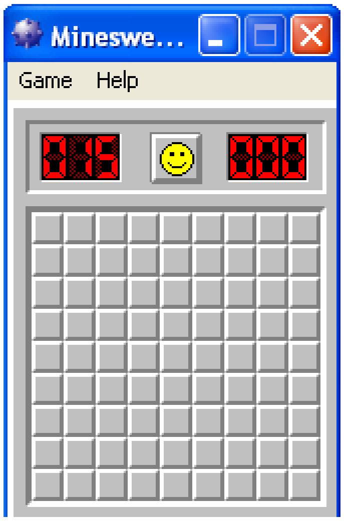
Recall the methods from the `Iterator` class.

| **Method Summary** | |
|---|---|
| `boolean` | **`hasNext`**`()`<br>Returns `true` if the iteration has more elements. |
| `Object` | **`next`**`()`<br>Returns the next element in the iteration. |
| `void` | **`remove`**`()`<br>Removes from the underlying collection the last element returned by the iterator. |

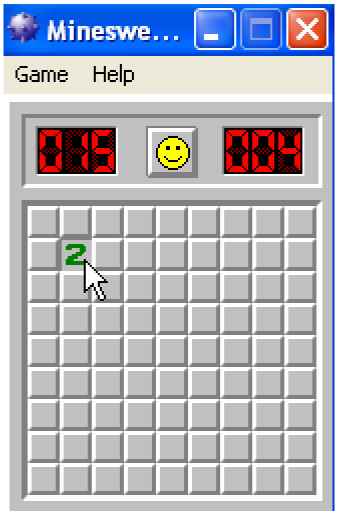
```
// pre: other != null
// post: neither this set or other is changed.
// returns a Cartesian Product of this set and other.
public ISet cartesianProduct(ISet other){
     assert other != null;
```

4. (Recursion, 20 points) Implement a method to update a Minesweeper board based on a cell that is selected.

In Minesweeper is a puzzle game that involves trying to deduce the location of mines on a board. The game takes place on a two dimensional array of cells.

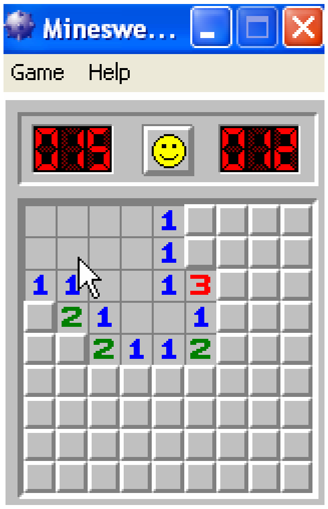


Each cell contains either a mine or a number that represents the number of mines that border that cell. In general each cell has eight neighbors (above, below, left, right, and the four diagonals) although there are special cases on the edges (5 neighbors) and corners (3 neighbors). When a cell is selected its contents are revealed. If it is a mine the game is over. If it is a positive int, meaning there are one or more mines bordering that cell, that number is revealed. For example when clicking the cell at row 1, column 1 I got the following result:



The 2 indicates that 2 of the 8 bordering cells have mines in them. The goal is to now select cells without picking a mine. In most versions of the game a shortcut is built in. If a cell is selected that has 0 bordering cells with mines, the game automatically reveals those bordering cells. And if any of those cells that were revealed have 0 bordering cells with mines, those bordering cells are revealed *and so forth...*

What happens when a cell that has zero mines bordering it is selected.



I restarted the game and clicked the cell at row 1, column 1. The location of the mines changes from game to game and in this instance there were 0 mines in the cells bordering cell (1, 1) so they were all revealed. Instead of showing a 0, this version of the game shows a blank. As it turns out some of the cells bordering (1,1) had 0 mines bordering them so all their neighbors were revealed. For example the cell at row 0, column 1. As you can see selecting one cell *might* result in many cells being revealed.

In this question you will only deal with one class, the `MinesweeperBoard` class. This class tracks the logical board as opposed the graphical board. The `MinesweeperBoard` class has two instance variables. A two dimensional array of ints that represents the true set up and a two dimensional array of booleans that indicates what cells are currently revealed to the player. Here is a simple example. Assume the board is 5 by 5. A -1 represents a mine. All cells that do not contain a mine are set to the number of mines bordering that cell.

| | | | | |
|---|---|---|---|---|
| **-1** | 2 | **-1** | 1 | 0 |
| 1 | 2 | 1 | 2 | 1 |
| 1 | 1 | 2 | 2 | **-1** |
| 1 | **-1** | 2 | **-1** | 2 |
| 1 | 1 | 2 | 1 | 1 |

There is a parallel two dimensional array of boolean. A cell is false if the actual contents of the cell are hidden from the player. A cell is true if the actual contents of the cell are being shown to the player. Initially all the elements of this two dimensional array are false.

| | | | | |
|---|---|---|---|---|
| false | false | false | false | false |
| false | false | false | false | false |
| false | false | false | false | false |
| false | false | false | false | false |
| false | false | false | false | false |

When a cell is selected the two dimensional array of booleans is updated. If a mine is chosen the game is over. If a cell with 1 or mines bordering it is selected that cell is revealed. If a cell that has 0 mines bordering it is selected that cell is revealed, as well as all of its neighbors. If any of those neighbors have 0 mines bordering them all of their neighbors are revealed and so forth.

Complete the update method for the `Board` class. If a mine is selected call the `endGame` method, otherwise set the appropriate cells in the `show` variable to true.

```
public class MinesweeperBoard{
     private static final int MINE = -1;
     private int[][] theTruth;
     private boolean[][] show;

     // call this method if a mine is selected.
     public void endgame()
     // do not implement this method

     // pre: none
     // Assume the player has selected a cell with coordinates
     // row and col. Update show appropriately as described in
     // the question.
     public void update(int row, int col){
          // complete this method
```

```
// more room on next page if necessary
// more room for update
```

Scratch Paper

Name:______________________________

TAs name: __________________________

Answer sheet for question 1, short answer questions

A. ______________________________

B. ______________________________

C. ______________________________

D. ______________________________

E. ______________________________

F. ______________________________

G. ______________________________

H. ______________________________

I. ______________________________

J. ______________________________

K. ______________________________

L. ______________________________

M. ______________________________

N. ______________________________

O. ______________________________