STUDENT NAME: _____

STUDENT ID: _____

McGill University
Faculty of Science
School of Computer Science

MIDTERM EXAMINATION

COMP-250: Introduction to Computer Science

October 22, 2010

Examiner: Prof. Doina Precup

Write your name at the top of this page. Answer directly on exam paper. Two blank pages are added at the end in case you need extra space. There are 3 questions worth 150 points in total. The last question also includes a bonus of 10 points. Partial credit will be given for incomplete or partially correct answers.

SUGGESTIONS: READ ALL THE QUESTIONS BEFORE YOU START!
GOOD LUCK!

1. [50 points] **Set difference**

   You are given two arrays, a of size n and b of size k. Each array contains distinct elements.

   (a) [35 points] Give an algorithm that returns a new array, c, containing the elements from a that are *not* contained in b. You can solve the problem in pseudocode or Java, whichever you prefer. You may assume that all algorithms we discussed in class this term are given to you, and you can call them at will.

   E.g. If array a is: 10 2 3 5 7 and array b is: 4 2 5, your algorithm should return array c containing: 10 3 7.

   (b) [15 points] Give the $O()$ of your algorithm in terms of n and k, and briefly explain how you came to your answer (e.g. "it's $O(n)$ because I have one loop that goes n times"). No formal proof is necessary.

2. [50 points] **Fixing bugs**

   The piece of Java code below is meant to find the maximum element in array *a*. There are 10 mistakes in the code. Indicate by comments at the side of the code what the mistakes are.

```
public class FindMax {
public void main(String[] args) {
int n;
int[] a = new int[n];

public static int max = 0;

for (i=1; i <= a.length; i++) {
if (a[i] > max) max = a[i]
}
return max;
}

}
```

3. [50 points + 10 points bonus] **Candy maze**

   You want to enter a Halloween contest in which you have to go through a triangle-shaped "maze" and gather as much candy as possible. From every room, there are two doors available. Once you go through a door, you cannot go back. Your friend who is helping set up the maze gives you the map of the maze ahead of time. You want to write an algorithm that computes the *maximum number of candies* that you can get.

   (a) [35 points] Suppose the maze is given to you as an array of size $n \times n$, where $a[i][j]$ is the number of candies in room (i, j). Only the elements below the diagonal are relevant. From $a[i][j]$, you can only go to $a[i+1][j]$ and $a[i+1][j+1]$. Write an algorithm that computes the maximum sum that can be obtained on any legal path through a.

   E.g. If a is the following array of size 1:
   2
   then there is only one path, of value 2.

   E.g. If a is the following array of size 2:
   2
   1 3
   then there are two paths:
   (2 1) of value 3
   (2 3) of value 5
   and you should return 5.

   E.g. If a is the following array of size 3:
   2
   1 3
   7 1 2
   then there are the following paths:
   (2 1 7) of value 10
   (2 1 1) of value 4
   (2 3 1) of value 6
   (2 3 2) of value 7
   and your algorithm should return 10.

   (b) [15 points] Prove that your algorithm finds correctly the path of maximum value, for any input.

   (c) [Extra credit: 10 points] Prove formally what is the $O()$ of your algorithm.

   Hint: Think recursion and induction.

Page left intentionally blank

Page left intentionally blank