CS/Math 240: Intro to Discrete Math	3/22/2011

Second Midterm Exam – Part I

Instructor: Dieter van Melkebeek

Guidelines:

- **Do NOT turn this page until you have received the signal to start. In the mean time read the instructions below carefully.**
- Fill in your name, student ID, and circle your discussion section below.
- This booklet consists of 6 sheets of paper, containing guidelines, 2 questions, and one extra credit question. When you receive the signal to start, please make sure that your copy of the test is complete.
- Do not separate the pages of this booklet.
- Answer each question directly on this booklet, in the space provided, and use the reverse side of the pages for rough work. If you need more space for one of your solutions, use the reverse side of a page and indicate clearly the part of your work that should be marked.
- In your answers, you may use without proof any result or theorem covered in lectures, lecture notes, discussion sections, or homework, as long as you give a clear statement of the result(s)/theorem(s) you are using. You must justify all other facts required for your solutions.
- Write up your solutions carefully! In particular, use notation and terminology correctly and explain what you are trying to do.
- Good luck!

Identifying Information:

Name:

Student ID:

Section:

301	302	303	304	305
Nilay	Nilay	Roger	Roger	Dalibor
M 9:55-10:45	W 9:55-10:45	M 12:05-12:55	W 12:05-12:55	W 1:20-2:10

For TA use only:

Score: / 10

Question 1: [5 points]

Consider the following program specification:

Input: An integer $n \geq 1$, an array $A[0..n-1]$ of integers that is *sorted* from smallest to largest, and an integer x.

Output: -1 if x does not appear in A, otherwise the first index i such that $A[i] = x$.

Consider the following implementation:

```
BinarySearch(n, A, x)
    i ← 0; j ← n − 1
    while i < j
        m ← ⌊(i + j)/2⌋
        if A[m] < x then i ← m + 1
                    else j ← m
    if A[i] = x then return i
                else return -1
```

1. State and prove an adequate *loop invariant*.

2. Use the loop invariant to prove *partial correctness*.

3. Specify and write an equivalent *recursive* program.

Question 2: [5 points]

For each of the following programs, prove or disprove that they correctly compute the greatest common divisor of positive integers a and b.

1.

```
GCD(a, b)
    if a = b then return a
    if a < b then return GCD(b − a, b)
              else return GCD(a, a − b)
```

2.

```
GCD(a, b)
    if a > b then a ← a − b
              else b ← b − a
    if a = b then return a
              else return GCD(a, b)
```

3.

GCD(a, b)

if $a = b$ **then return** a

if $a < b$ **then return** GCD$(a, b - a)$

else return GCD$(b, a(b + 1))$

4.

GCD(a, b)

if $a = b$ **then return** a

if $a < b$ **then return** GCD$(a, b - a)$

else return GCD(b, ab)

Extra Credit Question: [2 points]

For each of the programs in Question 2, determine exactly for which pairs (a, b) of positive integers the program correctly returns $\gcd(a, b)$.