CS/Math 240: Intro to Discrete Math 3/24/2011

Second Midterm Exam – Part II

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 5 sheets of paper, containing guidelines and 3 questions (numbered 3 through 5). 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 3: [2.5 points]

Let C_n denote the number of sequences of length n consisting only of 0's and 1's and containing no two consecutive 1's. For example, 00000, 01001, and 10101 are such sequences of length 5, whereas 00110, 11011, and 11111 are not.

1. Compute C_0 and C_1, and express C_n as a function of C_{n-1} and C_{n-2} for $n \geq 2$.

2. What is the solution to this recurrence?

Question 4: [5 points]

This problem deals with computing the Fibonacci numbers F_n.

1. Determine the exact number of additions the following program makes to compute F_n for $n \geq 2$.

```
Fib(n)
    F[1] ← 1; F[2] ← 1
    i ← 3
    while i ≤ n
        F[i] ← F[i − 1] + F[i − 2]
        i ← i + 1
    return F[n]
```

2. Do the same for the following program.

```
Fib(n)
    if n ≤ 2 then return 1
             else return Fib(n − 1) + Fib(n − 2)
```

3. The following identities hold for any integer $n \geq 2$:

$$\begin{aligned} F_{2n-1} &= F_n^2 + F_{n-1}^2 \\ F_{2n} &= (2F_{n-1} + F_n)F_n \end{aligned}$$

   Write an algorithm based on those identities that computes F_n using $O(\log n)$ integer additions and multiplications.
   Note: You do not need to prove the above identities.

Question 5: [2.5 points]

For each function $f : \mathbb{N} \to \mathbb{R}_{>0}$ in the left column, choose one expression $O(g)$ from the right column such that $f = O(g)$. Use each expression exactly once. You do not need to argue your answers.

(a)	$3 \cdot 2^n$	(i)	$O(\log n)$
(b)	$\frac{2n^4+1}{n^3+2n-1}$	(ii)	$O(n)$
(c)	$(n^5+7)(n^5-7)$	(iii)	$O(n \log n)$
(d)	$\frac{n^4 - n\log n}{n^2+1}$	(iv)	$O(n^2)$
(e)	$\frac{n \log n}{n-5}$	(v)	$O(n^{10})$
(f)	2^{3n+1}	(vi)	$O(2^n)$
(g)	$\prod_{i=1}^{n} i$	(vii)	$O(10^n)$
(h)	$(n-2)\log(n^3+4)$	(viii)	$O(n^n)$