ECE G205 Fundamentals of Computer Engineering
Fall 2004

Midterm Make-Up: Due by Wednesday December 1 2004

- This test is ONLY for those students that have gotten less than 80 (strictly less than 80) in the midterm. More specific instructions are given in the class web page.
- The test contains 4 problems. They allow you to earn 100 points.
- Show your work, as partial credit can be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. **Be neat**.
- **No late submissions will be accepted.**
- Only homework returned in a 9in $\times$ 12in envelope will be accepted. (If you cannot find such envelope, ask the Instructor.) Please, write your name and the class name (ECE G205) on the envelope (write clearly, please).
- For the problems below NO code has to be sent to the TAs.

Write your name here: ____

- **Problem # 1 [40 points]**. Consider the following 2×2 matrix:

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

(a) Write a function to *efficiently* compute the nth power of A, namely:

$$A^n = \underbrace{A \cdot A \cdots A}_{n \text{ times}}.$$

(A linear solution *won't* do ...)

(b) Determine the time complexity of your solution.

(c) How can a solution to point **(a)** be used to return the nth number of the Fibonacci sequence. (For a definition of the Fibonacci sequence, see Homework 7.)

- **Problem # 2 [20 points]**. As seen in class, the following is the recursive definition of the binomial coefficient when $k \leq n$

$$\binom{n}{k} = \begin{cases} 1 & \text{if } k = 0 \text{ or } k = n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{if } 0 < k < n. \end{cases}$$

Write a "Dynamic Programming" kind of function that returns $\binom{n}{k}$. As mentioned, the binomial coefficient $\binom{n}{k}$ can be seen as the last entry $c_{n,k}$ of a $(n+1) \times (k+1)$ matrix which can be filled out line by line in the following way: $c_{0,j} = 0$, $1 \leq j \leq k$, $c_{i,0} = 1$, $0 \leq i \leq n$, and

$$c_{i,j} = c_{i-1,j-1} + c_{i-1,j}, 1 \leq i \leq n, 1 \leq j \leq k.$$

Use this characterization (also known as *Pascal's triangle*) to write a function that taken as input n and k returns $\binom{n}{k}$. The *space* complexity of your solution *MUST* be linear, i.e., $O(n)$.

- **Problem # 3 [30 points]**. Write a Dynamic Programming algorithm to compute a^n (n even) based on the formula:

$$a^n = a^{n/2}a^{n/2}$$

- **Problem # 4 [30 points]**. In case of small x and y write a C++ function that returns x^y in constant time.