Computer Science 364 | Summer, 2003
St. George Campus | University of Toronto

Homework Assignment #2
Due: July 2, 2003, by 6pm
(in the lecture)

On the cover page of your assignment, you must write **and sign** *the following statement: "I have read and understood the homework collaboration policy described on the course webpage." Without such a signed statement your homework will not be marked.*

---

[20] 1. **Offering courses**
In a certain college, there are n instructors and m different courses that the college can offer. For every instructor i, there is a list of courses $C_i \subseteq \{1, \ldots, m\}$ that this instructor can teach. In one semester, an instructor can teach no more than three courses.

[10] (a) Given the n, m and C_i for $1 \leq i \leq n$, determine the maximal number of different courses the college can offer in one semester. Hint: design a flow network that represents this problem.

[5] (b) Give an example of an input (n, m and all C_i) and the corresponding network where the college cannot offer all courses, although there is somebody who can teach each course.

[5] (c) How can you recover the resulting assignment of instructors to courses given the maximum flow on your flow network?

[10] 2. **Divisibility by three**
Give a Turing machine that, on a binary input x, determines whether x is divisible by three. Give a state table with comments describing clearly what every state is doing. Your machine should not have more than 10 states. Show how your machine works on a small example.

[20] 3. **s-t LONGEST-PATH**
An s-t LONGEST-PATH search problem (STLPS) is: given a graph G, and two nodes s and t, find a longest simple path from s to t, if one exists; output "no path" otherwise.

[5] (a) Define a corresponding decision problem STLPD.

[5] (b) Show how to compute the length of the longest path from s to t in G by reducing this problem to STLPD. If there is no path, output -1. Your reduction should run in time $O(\log n)$.

[10] (c) Show how to solve STLPS by reducing it to STLPD.

[15] 4. **Concatenation of NP languages.**
Suppose that both L_1 and L_2 are in NP. Define a concatenation of L_1 and L_2 as $L_3 = \{x_1 x_2 | x_1 \in L_1 \text{ and } x_2 \in L_2\}$ (that is, strings in L_3 are of the form a string from L_1 concatenated with a string of L_2.) Show using the formal definition of NP that L_3 is in NP as well.