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

Homework Assignment #2
Due: Oct 29, 2003, by 2pm
(in the tutorial)

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. **Network flows: Most profitable recording**

A performer wants to record some of her compositions. For that, she needs to hire a group of musicians. Not all of her compositions require the same sets of instruments: some can be played with just an acoustic guitar, some need a harpsichord or a sitar.

Assume that each musician that she hires can play exactly one instrument. Once a musician is hired, for a fixed sum, he agrees to play in all compositions that require his instrument. Each composition has a gain associated with it: it is, loosely, the expected sum of money that putting this piece on the recording will bring; some pieces are better than others. There is no limit on the number of pieces to be recorded, but she would not want to hire a really expensive musician to record a mediocre piece with gain that is not going to even cover that musician's salary.

Suppose that there are n pieces, where i^{th} piece has an associated list M_i of musicians needed to record it and an expected gain g_i. There are also m musicians, where j^{th} musician can be hired for a cost c_j. For example, there can be 3 pieces with expected gains $5, 30$ and 10, and two musicians (say a guitarist and a pianist), where the costs are 22 to hire the guitarist and 17 to hire the pianist. Suppose that the first and second pieces need just the guitarist and the last one needs both. Then it is more profitable to hire only the guitarist and record the first two pieces, than having to pay 17 to the pianist for the increase in gain of only 10. Thus, the most profitable recording will consist of the first two pieces, and it will bring the net profit of $30 + 5 - 22 = 13$.

We will solve this problem using the approach similar to the maximum matching problem. Define a flow network as follows. The set of vertices in G consists of special vertices s and t, together with n vertices corresponding to pieces and m vertices corresponding to musicians. The set of edges in G is as follows: 1) For every piece i, there is an edge (s, i) with capacity g_i. 2) For every musician j there is an edge (j, t) with capacity c_j. 3) If i is a piece and j is a musician such that $j \in M_i$, then an edge (i, j) with infinite capacity.

[5] (a) Show that a cut (S, T) in G has **finite** capacity iff for every piece $i \in S$, if $j \in M_i$ then $j \in S$. That is, if a piece is in S, then all musicians it needs are also in S.

[3] (b) Give an example of a set of pieces and musicians, a corresponding graph, and a cut in that graph that corresponds to a recording where for all pieces there are necessary musicians, but there are extra musicians hired and a piece that has all necessary musicians hired is not included.

[7] (c) Show how to express the net profit of a recording as a function of the capacity of a corresponding cut and sums of all gains and all costs.

[5] (d) State how, given the graph, you can determine which pieces and which musicians are chosen for a best recording and what is the net profit of that recording. Can there be several best recordings?

[20] 2. **Turing machines: one-way infinite tape**
Suppose that we change the definition of a Turing machine to have a tape that is only infinite in one direction (say, infinite to the left). That is, there is a first cell of the tape, and if the machine tries to go right from it, the head stays at the same place. Assume also that the input starts from the first cell, so at the first step the head always points to the first cell. Prove that these two definitions are equivalent.

[10] (a) Show how to simulate a one-way tape Turing machine by a two-way tape Turing machine. That is, given a one-way tape machine $M = \{\Sigma, \Gamma, Q, \delta\}$, describe an equivalent two-way machine $M' = \{\Sigma, \Gamma', Q', \delta'\}$. State which modifications to Γ, Q and δ are necessary.

[10] (b) Show how to simulate a two-way tape Turing machine by a one-way tape Turing machine. The proof is similar to above: given a two-way tape machine M describe an equivalent one-way tape machine M'.

[20] 3. **Search-to-decision reduction: Clique search problem**
Let G be an undirected graph. A *clique* in a graph is a complete subgraph, that is, a subset of vertices $S \subseteq V$ such that for all $i, j \in S$, $(i, j) \in E$. A Clique search problem is: given a graph G, find the largest clique in G, that is, $S \subseteq V$ such that S is a clique in G and for any $S' \subseteq V$ if S' has more vertices than S then S' is not a clique in G.

[2] (a) Give an example of a graph on 5 vertices that has a clique of size 4, but not of size 5.

[4] (b) Define a decision problem CliqueD that you can use to solve Clique search problem.

[7] (c) Let CliqueSize be the problem of computing the *size* of the largest clique in G. State how to solve CliqueSize by reducing it to CliqueD. Your reduction should be polytime and make no more than $O(\log n)$ calls to CliqueD, where n is the number of vertices in G.

[7] (d) Now solve the Clique search problem by reducing it to CliqueD. You can call CliqueSize inside your algorithm. Your reduction should be polytime, in particular it should make no more than $O(n)$ calls to CliqueD.

[20] 4. **Definition of NP: sum of binary numbers**
Let $L_1 \subseteq \{0,1\}^*$ and $L_2 \subseteq \{0,1\}^*$ be two languages in NP. Define a language $L_3 = \{x_1 + x_2 \mid x_1 \in L_1 \text{ and } x_2 \in L_2\}$. That is, L_3 contains all binary strings that correspond to binary numbers representable as a sum of a binary number from L_1 and a binary number from L_2. Prove, *using the definition of NP*, that L_3 is in NP. Hint: given x, how do you witness that $x \in L_3$?