CSI 4105 Design and Analysis of Algorithms II — Fall 2002
Computer Science — University of Ottawa

Homework Assignment #2 (100 points, weight 10%)
Due: Thursday, November 14, at 10:00 a.m. (in lecture)

The independent-set problem and an application.

You were hired by the Athletics facilities at Y University which is holding a sports day for new students and need to schedule one game for each sport modality. Each student can sign up for one or more modalities. Games can be scheduled simultaneously, as long as no student has signed up for two of them. Space to hold the games simultaneously is not an issue at Y University.
There is a very special timeslot on Saturday afternoon at 2:00 p.m. in which the university officials would like to have as many games going on as possible. At this time slot, there will be an open house for prospects sponsors for sports events, and the university would like to have its courts full. Your job is to schedule as many games as possible for the timeslot on Saturday at 2:00 pm.
Your boss has studied algorithms, and briefly told you:

> All you have to do is to model the problem as a maximum independent set in a graph. The graph has one vertex for each game; two vertices are connected by an edge whenever at least one student has signed up for both games corresponding to the two vertices. Scheduling the maximum number of games for the Saturday open house corresponds to finding the maximum independent set in this graph. A co-op student has already built the graph based on the student sign up sheets; you can concentrate on the independent-set algorithm.

After that, you say to your boss: "Oh Yes, of course." Then, you go back to your "Introduction to Algorithms" book by Cormen, Leiserson and Rivest to study the definitions for independent sets...

An *independent set* in a graph $G = (V, E)$ is a subset $I \subseteq V$ such that for all $x, y \in I$, $\{x, y\} \notin E$.
The independent-set problem consists of finding a maximum (largest) independent set in a graph. This optimization problem can be transformed into the decision problem corresponding to the following language:

$$\textsc{IndSet} = \{< G, k >: G \text{ is a graph with an independent set of size } k\}$$

Then you recall:
"Oh. I have shown in the CSI 4105 midterm test that IndSet is NP-complete. I even recall that I used a reduction from the Clique problem."

You then start investigating several aspects of the problem:

1. (10 points) **Understanding the application.**
Create a small example of the student sign up sheets for 5 games. Build a graph based on the sign up sheets and indicate a maximum independent set in your graph. Write the schedule of games for the open house time slot. You have to limit all this to 1 page maximum!!!

2. (30 points) **Designing a greedy heuristic algorithm.**

- (25 points) Design a polynomial time greedy heuristic algorithm to find a large independent set. The greedy heuristic should consist of selecting one vertex at a time for the independent set, according to some criteria you choose. Note that the final solution should be an independent set, even though it may not be the largest possible. Please, briefly explain any intuition behind your heuristic. Briefly justify that your algorithm runs in polynomial time.
- (5 points) Give an example of a graph for which your algorithm constructs an independent set with size smaller than the maximum one. One such example should exist, unless your polynomial time algorithm in the previous part always find the largest independent set (in this case you have proven that P=NP!).

3. (30 points) **Polytime decision implies polytime optimization for this problem.**
Suppose you are given a "black-box" subroutine to solve the decision problem associated to IndSet. Give an algorithm to find an independent set of maximum size. The running time of your algorithm should be polynomial in $|V|$ and $|E|$, where queries to the black box are counted as a single step; show that the algorithm runs in polynomial time in $|V|$ and $|E|$.

4. (30 points) **Polynomial-time algorithm for a special case.**
You have noticed that in the graph you have, each game conflicts with exactly two other games, and decided to investigate whether an efficient algorithm is available for this type of graph...
Give a polynomial time algorithm to find the maximum independent-set in a graph G when each vertex of G has degree 2. Prove that your algorithm works correctly. Analyse its running time.

NP-completeness proof. (Practice exercise, optional)
You may need more practice with NP-completeness proofs. This exercise is optional, just for the sake of practice, but will be marked by the TA if you hand it in.
5. (0 points) Let

$$\text{HALFCLIQUE} = \{< G > \; : \; G \text{ is a graph having a clique with at least } n/2 \text{ vertices, where } n \text{ is the number of vertices in } G\}$$

Hint: Reduce from Clique; you may add vertices/edges to the graph instance of Clique.