Due: By 6:00pm on Thursday 7 December, in the CSC 373H drop box in room BA 2220. **Worth**: 7.5%

1. [20 marks] Do Problem 17 in Chapter 7 in the book (network diagnosis). (Hint: try to find the corrupt edges.)

2. [25 marks] Do Problem 20 in Chapter 7 in the book (baloons).

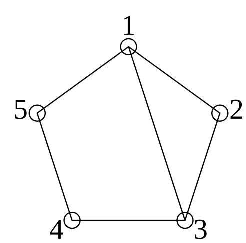
3. [8 marks] Show how to formulate the problems on this page as $\{0, 1\}$ integer programming (IP) problems.

   (a) [5 marks] The MaxSat problem: the input is a propositional formula in CNF, $F = C_1 \wedge C_2 \wedge \cdots \wedge C_r$, where each clause is a disjunction of one or more literals (propositional variables or negated propositional variables) and we want to know the maximum number of clauses that can be satisfied simultaneously by a single truth assignment. (*Hint:* If there are r clauses and n propositional variables in F, then it suffices to have $r + n$ variables in the IP.)

   (b) [3 marks] The scheduling problem of maximizing the profit of scheduled activities $A_i = (s_i, f_i, p_i)$ where $s_i \geq 0$ is the start time, $f_i > s_i$ is the finish time, and $p_i \geq 0$ is the profit of activity A_i, for $i = 1, 2, \ldots, n$. The activities must be scheduled on one machine without overlap.

4. [15 marks]

   Consider the *Unweighted Maximum Cut* problem: given an undirected graph $G = (V, E)$, find a partition (A, B) of V so as to maximize $|cut(A, B)|$ where $cut(A, B)$ is the set of edges with one endpoint in A and the other endpoint in B. For convenience, let $V = \{1, 2, \ldots, n\}$.
   For example, in the graph G pictured on the right, $A = \{1, 4, 5\}$, $B = \{2, 3\}$ is a cut of size 3 because $cut(A, B) = \big\{(1, 2), (1, 3), (4, 3)\big\}$, and $A' = \{3, 5\}$, $B' = \{1, 2, 4\}$ is a cut of size 5 because $cut(A', B') = \big\{(3, 1), (3, 2), (3, 4), (5, 1), (5, 4)\big\}$.



   Consider the following greedy algorithm for the Unweighted Maximum Cut problem:

   $A := \emptyset$
   $B := \emptyset$
   for $v := 1, 2, \ldots, n$:
       $a := \big|\{(u, v) : u \in B\}\big|$ // number of new edges cut if v is put into A (0 if $B = \emptyset$)
       $b := \big|\{(u, v) : u \in A\}\big|$ // number of new edges cut if v is put into B (0 if $A = \emptyset$)
       if $a > b$: // choose the option the maximizes the number of new edges cut
           $A := A \cup \{v\}$
       else:
           $B := B \cup \{v\}$

   For example, on the graph G pictured above, the algorithm produces the cut $A = \{2, 4\}$, $B = \{1, 3, 5\}$.

   (a) [6 marks] When the algorithm terminates on an arbitrary graph G, for every vertex $v \in V$, let $cut(v)$ be the set of edges (u, v) crossing the cut such that $u < v$, and let $uncut(v)$ be the set of edges (u, v) **not** crossing the cut such that $u < v$. (For example, for the graph G pictured above, $cut(3) = \big\{(3, 2)\big\}$ and $uncut(3) = \big\{(3, 1)\big\}$.)
   Prove that for any graph G, for every vertex $v \in V$, $\big|cut(v)\big| \geq \big|uncut(v)\big|$.

(b) [7 marks] Prove that upon termination of the algorithm on an arbitrary input G, at least half of all the edges are in $cut(A, B)$. (You may use the result from the previous part even if you did not prove it.)

(c) [2 marks] Prove that upon termination of the algorithm on an arbitrary input G, $|cut(A, B)| \geq \frac{1}{2}OPT$, where OPT is the maximum number of edges crossing any cut. (You may use the result from the previous parts even if you did not prove it.)