CS 341, Fall 2011
Solutions for Quiz 2, Day Section

1. (a) There are two different approaches one can use to show that $ILP \in \mathrm{NP}$:

   - Show that ILP has a nondeterministic Turing machine that runs in polynomial time, or
   - Show that ILP has a polynomial-time verifier.

   We will use the second approach. We now give a polynomial-time verifier using as a certificate an integer vector c such that $Ac \leq b$. Here is a verifier for ILP:

   V = "On input $\langle\langle A, b\rangle, c\rangle$:
      1. Test whether c is a vector of all integers.
      2. Test whether $Ac \leq b$.
      3. If both tests pass, *accept*; otherwise, *reject*."

   If $Ay \leq b$ has m inequalities and n variables, then Stage 1 takes $O(n)$ time, and Stage 2 takes $O(mn)$ time. Hence, verifier V has $O(mn)$ running time, which is polynomial in size of problem.

   (b) Now we show $3SAT \leq_{\mathrm{m}} ILP$. (We later show the reduction takes polynomial time.) To do this, we need an algorithm that takes any instance ϕ of the $3SAT$ problem and converts it into an instance of the ILP problem such that $\langle\phi\rangle \in 3SAT$ if and only if the construct integer linear program has an integer solution. Suppose that ϕ has k clauses and m variables $x_1, x_2, \ldots, x_m$. For the integer linear program, define $2m$ variables $y_1, y_1', y_2, y_2', \ldots, y_m, y_m'$. Each y_i corresponds to x_i, and each y_i' corresponds to $\overline{x_i}$. For each $i = 1, 2, \ldots, m$, define the following inequality and equality relations to be satisfied in the integer linear program:

   $$0 \leq y_i \leq 1, \quad 0 \leq y_i' \leq 1, \quad y_i + y_i' = 1. \tag{1}$$

   If y_i must be integer-valued and $0 \leq y_i \leq 1$, then we must have y_i can only take on the value 0 or 1. Similarly, y_i' can only take on the value 0 or 1. Hence, $y_i + y_i' = 1$ ensures exactly one of y_i, y_i' is 1 and the other is 0. This corresponds exactly to what x_i and $\overline{x_i}$ must satisfy.

   Each clause in ϕ has the form $(x_i \vee \overline{x_j} \vee x_k)$. For each such clause, create a corresponding inequality

   $$y_i + y_j' + y_k \geq 1 \tag{2}$$

   to be included in the integer linear program. This ensures that each clause at least one true literal. By construction, ϕ is satisfiable if and only if the constructed integer linear program with m sets of relations in display (1) and k inequations as in display (2) has an integer solution. Hence, we have shown $3SAT \leq_{\mathrm{m}} ILP$.

   (c) Now we have to show that the time to construct the integer linear program from a 3cnf-function ϕ is polynomial in the size of $\langle\phi\rangle$, which we can measure in terms of the number m of variables and the number k of clauses in ϕ. For each $i = 1, 2, \ldots, m$, display (1) comprises 6 inequalities $y_i \geq 0$ (rewritten as $-y_i \leq 0$),

$y_i \leq 1$, $y_i' \geq 0$ (rewritten as $-y_i' \leq 0$), $y_i' \leq 1$, $y_i + y_i' \leq 1$, and $y_i + y_i' \geq 1$ (rewritten as $-y_i - y_i' \leq -1$), where the last two together are equivalent to $y_i + y_i' = 1$. Thus, we have $6m$ inequalities corresponding to display (1). The k clauses in ϕ leads to k more inequalities, each of the form in display (2). Thus, the constructed integer linear program has $2m$ variables and $6m + k$ linear inequalities, so the size of the resulting integer linear program is polynomial in m and k. Hence, the reduction takes polynomial time.