15-780 Homework 4
Deadline: 10:30 am on April 2 (Thursday)
There are 100 total points: point values are listed with each question.

1. **Planning (50 pts)**

   Congratulations, your class project has just been accepted at a prestigious conference to be held in Tokyo this summer! As a conscientious student of grad AI, you of course decide to model your problem of attending the conference as a planning problem. The Conference Attendance planning domain is defined as follows:

   - A grad student starts at Pittsburgh.
   - The grad student must travel to Tokyo, attend the conference, and return to Pittsburgh.

   We recommend a semi-automated approach to answering the following questions: you may feel free to use any combination of manual and code-based construction of the plan graph and SAT problem. Please hand in any code that you do use, along with the manually-entered data which it works from.

   (a) Write the planning domain above using a STRIPS-like language. Recall that this involves specifying a set of state predicates and actions (an example is given in the lecture notes). Be sure to describe the pre-conditions and post-conditions of all of the actions you listed. If your representation contains more than 2–3 distinct operators, you are probably trying to put in more detail than is necessary for this problem. Also specify the initial state and goal predicate. (10 pts)

   (b) Draw the plan graph that represents the task of attending the conference as described above (**you need only draw enough levels to reach the first appearance of all of the goal predicates**). Separately, list (or print out) all mutex links (these links should *not* be added to the graph). Please separate your list by type of mutex: contradictory post-conditions, interference with pre-conditions, mutex pre-conditions, and literals that can't be achieved by a pair of non-mutex actions. You need not explicitly include mutexes between contradictory literals such as x and $\neg x$, but you should of course consider these mutexes when deriving mutexes at the next level. (20 pts)

   (c) Using the techniques described in class (and in RN p. 402), take the plan graph of Problem 1b and write it as a SAT problem in CNF. Note: if we deducted points for your plan-graph in Problem 1b, you will not be further penalized in this problem as long as you correctly translate the graph you showed. (10 pts)

   (d) Use your plan graph to find a solution. Recall that you are free to use any combination of manual and automated methods: for example, you could run your SAT solver from previous homeworks on the formula you derived in part 1c (or you could use the plan graph algorithm described in lecture). Write your solution as a partially-ordered list of actions. How many operators did it take to solve this problem? How many levels did you need in the plan graph? Did you need to extend your plan graph farther than required by 1b? (Recall that the algorithm for finding plans from a plan graph dictates that, if we fail to find a solution using a graph with a given number of levels, we should add another level and try again.) (10 pts)

2. **Linear and Integer Programming (50 pts)**

Consider the standard-form LP problem, $x \in \mathbb{R}^n$:

$$\text{minimize (or maximize):}$$

$$c^T x$$

$$\text{subject to:}$$

$$Ax = b$$
$$x \geq 0$$

where A is an $m \times n$ matrix, with $n \geq m$. The simplex method is an iterative algorithm for solving this LP (geometrically, it travels from one vertex to another until a local minimum is found).

At each iteration, the simplex algorithm returns a candidate solution $x \in \mathbb{R}^n$. The vector x has the property that at least $n - m$ of its components are zero. Let N denote a set of $n - m$ indices of x that are zero, and let B denote the remaining variables. Variables with indices in B are called *basic variables*, and variables with indices in N are called *nonbasic variables.* Let x_B denote the vector of basic variables, and x_N denote the vector of nonbasic variables. Furthermore, let us assume that the first m components of x are the basic variables (i.e., that $x = [x_B; x_N]$). We can assume this without loss of generality, since otherwise we could just permute the elements of x and the corresponding columns of A.

Now we partition A into two components. The first will be an $m \times m$ matrix consisting of the first m rows and the first m columns of A. The simplex algorithm will only produce bases such that this first block of A is invertible; we will call it S^{-1}. Let C denote the remaining $m \times (n - m)$ submatrix of A, so that $A = [S^{-1}\ C]$, where S is a square $m \times m$ matrix.

Thus, we can rewrite our original system of constraints $Ax = b, x \geq 0$ equivalently as

$$[I_m\ SC][x_B; x_N] = Sb,\ x \geq 0,$$

where I_m is the $m \times m$ identity matrix. This system of equations is called the *simplex tableau.*

(a) Consider the following linear program:

$$\text{maximize:}$$

$$3x + 2y$$

$$\text{subject to:}$$

$$2x + y \leq 10$$
$$x + 3y \leq 12$$
$$x, y \geq 0$$

What is the optimal solution of this LP? Please find this by hand by sketching the feasible region and determining the intersection of the active constraints. (10 pts)

(b) Derive the dual of this linear program by hand. (10 pts)

(c) Now suppose we add the additional constraints that x and y are integers. We add two slack variables s_1 and s_2 to put the LP into standard form, and then turn it into an ILP by constraining all variables to be integral.

maximize:

$$3x + 2y$$

subject to:

$$\begin{aligned} 2x + y + s_1 &= 10 \\ x + 3y + s_2 &= 12 \\ x, y, s_1, s_2 &\geq 0 \text{ and integer} \end{aligned}$$

Consider the LP relaxation of this ILP (which is obtained by dropping the integrality constraints). Create the simplex tableau with basis $B = \{x, y\}$, using the procedure described above. (10 pts)

(d) Recall the Gomory cut algorithm described in lecture. Given a row from the simplex tableau, this algorithm outputs an additional constraint known as a *cut*, which eliminates feasible solutions from the LP relaxation without eliminating any integral points.

Please run the Gomory cut algorithm by hand using the row from your simplex tableau corresponding to basic variable x. What is the resulting cut? Sketch the feasible region from part (a) again, including the new cut you added. Does the new cut result in an integer optimal solution. Will this always be the case for a Gomory cut? What about other families of cuts—in general, if procedure A always returns a cut which results in an integer optimal solution, what can you say about the runtime of procedure A (under any standard assumptions you wish to make)? (20 pts)