You can look up material on the web and books, but you cannot look up solutions to the given problems. You can work in groups, but must write up the answers individually.

Problem 1: Testing for Nesting of Polytopes (10pt)
Let $S = \{x \in R^n : Ax \leq b\}$ and $T = \{x \in R^n : Bx \leq d\}$. Assuming that S and T are nonempty, describe a polynomial time algorithm (in n) for checking whether $S \subseteq T$.

Problem 2: Max Flow Dual (10 points)
The class slides described a formulation of Max-Flow using linear programming (Page 7 of the first lecture on linear programming). Assume there is no capacity inequality for the special edge x_0 (the edge from out to in can support any capacity). The same lecture also described how to convert a linear program into its dual (page 44). Please write down the dual for the max-flow problem and write down an interpretation of what the equations mean, what the variables mean, what we are optimizing, what the optimal solution will look like, and why these particular equations optimize what you claim they do. I'm not looking for an essay—short answers are best.

Problem 3: Approximation Algorithms using Linear Programming (10 points)
By rounding a linear solution to an integer solution it is sometimes possible to get good approximations to various NP-hard problems (e.g. an answer that is within a constant factor of optimal). This is an approach that has been used extensively in the past 10 years to get good approximation bounds on various hard problems. Here is an example.
The *Set Cover* problem consists of a set of elements U and a family of subsets $\{S_1, S_2, \ldots, S_m\}$ with each $S_i \subseteq U$ and $\cup_i S_i = U$. Each set S_i has a cost C_i. We consider special instances of Set Cover, where each element is contained in at most f sets. This problem is still NP-hard, so we don't know how to solve the integer linear program formulation below in polynomial time.

$$\min \textstyle\sum_i C_i x_i \tag{1}$$
$$\text{subject to } \textstyle\sum_{i:j \in S_i} x_i \geq 1 \qquad \text{for all elements } j \in U \tag{2}$$
$$x_i \in \{0, 1\}. \tag{3}$$

So instead, you solve the *LP relaxation* obtained by replacing the integrality constraints (3) by non-negativity constraints.

$$\min \textstyle\sum_i C_i x_i \tag{4}$$
$$\text{subject to } \textstyle\sum_{i:j \in S_i} x_i \geq 1 \qquad \text{for all elements } j \in U \tag{5}$$
$$x_i \geq 0. \tag{6}$$

Let Z_{IP} be the optimal value of the integer program, Z_{LP} be the optimal value of the LP relaxation (with the values of the variables in the LP being x_i^*).

1. Show that $Z_{LP} \leq Z_{IP}$.
2. Given a positive value τ, consider the following *rounding* operation: set $x'_i = 1$ if $x^*_i \geq \tau$, and $x'_i = 0$ otherwise. What is the largest value of τ (in terms of f) which will always ensure that x' is a feasible solution to the integer program.
3. Show that the cost of the solution $\sum_i C_i x'_i \leq (1/\tau) \times Z_{IP}$, and hence we have a set cover solution whose cost is at most $1/\tau$ times the cost of the optimal solution.

Problem 4: Branch and Bound (10 points)

Solve the following problem using the implicit enumeration branch-and-bound technique (the 0-1 programming technique) describe in class. You should show the tree and mark on any leaf node why it was pruned.

$$\begin{array}{ll} \text{minimize} & 3x_1 + 2x_2 + 5x_3 + x_4 \\ \text{subject to} & -2x_1 + x_2 - x_3 - 2x_4 \leq -2 \\ & -x_1 - 5x_2 - 2x_3 + 3x_4 \leq -3 \\ & x_1, x_2, x_3, x_4 \text{ binary} \end{array}$$