September 7th and September 11th Lectures

We saw how to formulate some further problems as linear programs or integer linear programs. The formulation are given in the powerpoints. They include Exercise 1.6 of Chvatal which involved the choice of what meat products a smoker should be used to smoke, and Exercise 1.8 which involved scheduling assembly and related training for a contract for assembling 20,000 radios. These are simple toy examples which give a flavor of the sophisticated resource allocation/scheduling problems which are solved by businesses after having been formulated as LPs or ILPs. These routinely have thousands of constraints and variables.

We saw how to formulate three problems you have seen in earlier courses as LPs or ILPs. Specifically we saw how to formulate Minimum Capacity s-t Cut, and Minimum Weight Spanning Tree as Integer Linear Programs. In formulating MWST, we exploited the fact set out, e.g. on page 624 of Cormen et al, that T is a spanning tree of G precisely if it contains |V(G)-1| edges and is acyclic. To ensure acyclicity we added a constraint for each cycle C of G. We also formulated Maximum Volume s-t Flow as an LP and noted that if we want an integer valued flow we could simply add the constraint that the variables be integers and obtain an ILP.
We noted that all of these problems could be solved more quickly using the specialized algorithms you have seen in earlier courses, including the Integer Max-Flow problem.

We also discussed how to formulate variants of the knapsack problem as ILPs or LPs. In these problems, we are choosing a subset of a set of items, an item of type I has a weight w_i and a value v_i . We have a bound W on the overall weight that the knapsack can hold, and we want to maximize the sum of the values that we put into it. In a vanilla knapsack problem, there is only one item of each time, so the variable x_i setting out how many items of type i we put in the knapsack is 0-1 valued. In the multi-knapsack problem, we are in a store with an essentially unlimited number of each item, so x_i is simply restricted to be a nonnegative integer. In the fractional versions of these problems, we drop the integrality constraints. (See Powerpoints)

We also showed how to formulate Satisfiability as an ILP. (See powerpoints)

Finally we discussed matrix games, using *Rock, Paper, Scissors*, and *Morra* as our two examples. Our discussion of the Morra example draws on Chapter 15 of Chvatal which is posted in Mycourses. Note you need only look at pages 228-233 (stop at *The Minimax Theorem* header). Furthermore, although Chvatal mentions Matrix notation and uses the notation xAy, you can essentially ignore this. The rest of what he says makes sense without it.