1. 360 students have preenrolled for CS 61B. Each 61B student must attend one of the twelve lab sections, and each lab section can hold at most 30 students. Each student has specified three choices for lab section. A "happiness" rating is assigned to each choice: assignment of a student to his or her first choice is worth 10, while the second choice is worth 7, and the third choice 3. Assignment of a student to a section not among his or her top three is worth 0. The problem is to determine an assignment of students to lab sections that maximizes the total happiness of the enrolled students, without exceeding the capacity of any lab section.

   Express this problem as

   - a linear programming problem, and
   - a maximum flow problem.

2. CLR Exercise 27.2-9.

3. CLR Exercise 27.3-1.

4. *Generalizations of the max-flow problem.* The max-flow problem can be generalized in many different directions:

   - There are many sources and many sinks, and we wish to maximize the total flow from all sources to all sinks.
   - Each edge has not only a capacity, but also a *lower bound* on the flow it must carry.
   - The outgoing flow from each node v is not the same as the incoming flow, but is smaller by a factor of $(1 - \epsilon_v)$, where ϵ_v is a loss coefficient associated with node v.
   - Each edge has a cost per unit flow associated with it, and we must find among all flows of maximum value the one that minimizes the total cost.

   In each case, show how to solve the more general problem by reducing it to an ordinary max-flow problem when possible; or by reducing it to a linear program if you cannot reduce it to a max-flow problem.

5. One graph G is defined to be *isomorphic* to another graph H when the vertices of G can be relabeled in such a way that a copy of H results. We'll call this the "graph isomorphism problem." (A more formal definition of graph isomorphism appears in CLR, page 88.) Interestingly, it is not known if there is a polynomial-time lgorithm for determining whether one graph is isomorphic to another, nor has the problem been proven to be NP-hard.

   - Is the graph isomorphism problem in NP? Explain.
   - A superficially similar problem, determining whether a subgraph of a given graph G is isomorphic to another graph H, is NP-hard. We'll call this the "subgraph isomorphism problem." Is the graph isomorphism problem reducible to the subgraph isomorphism problem? Explain.
   - Is the subgraph isomorphism problem reducible to the graph isomorphism problem? Explain.

6. CLR Exercise 36.2-3.
7. CLR Exercise 36.4-6.