CSEP 521 - Spring 2005
Assignment 3

Due 4/21/05

1. A number of families are planning a driving vacation together. In order to avoid turmoil it was decided that no two members of the same family should travel in the same car. Assume there are n families where family i has size $s_i \geq 1$ and car with capacity $c_i \geq 1$.

   (a) Formulate this problem as a network flow problem assuming that any person can drive any car. A solution would be an assignment of people to cars with the property that no car has two members from the same family. Not all cars need be used.

   (b) Reformulate the problem assuming that exactly one person from each family drives its own car and all cars are used for the trip.

2. The max flow, min cost problem has a polynomial time solution and many applications. The problem can be stated succinctly by:

   - Input: A directed graph $G = (V, E)$ where each edge $(u, v) \in E$ has an associated cost, cost(u, v) and capacity $c(u, v)$.
   - Output: A maximum flow of minimal cost. That is a legal flow f such that f is a maximum flow and $\sum_{f(u,v)>0}$ cost$(u, v) f(u, v)$ is minimized.

   Suppose we have the following assignment problem of assigning bank customers to tellers. There are n customers and m tellers, each of whom speaks one or more languages. We would like to match each customer to a teller who speaks the same language so as to minimize the overall waiting time for all the customers. Suppose a total $i > 0$ customers are assigned to the same teller, then the waiting time for the first customer is 0, for the second customer is 1, for the third customer is 2, and so on. The waiting time for all the i customers in this line is $0 + 1 + 2 + \cdots i - 1 = (i - 1)i/2$.

   This kind of problem can be formulated as a *semi-matching problem*, where the input is a bipartite graph $G = (U, V, E)$ (Recall the $E \subseteq U \times V$.) The output is a semi-matching, that is, a subset M of E with the property that $(u, v), (u, w) \in M$ implies $v = w$ and for all $u \in U$ there is a $v \in V$ such that $(u, v) \in M$. That is, each member of U is matched with exactly one member of V, but not vice versa. The optimization goal in semi-matching is to find a semi-matching M which minimizes $\sum_{v \in V}(m_v - 1)m_v/2$ where m_v is the number of members of U that are matched with v.

   Given an instance of the semi-matching problem, show how to construct an instance of the max flow, min cost problem in polynomial time, that solves the semi-matching problem.