CSE 521
Assignment 2
Due Tuesday, April 15, 2003

1. In this problem we consider the case were we are trying to find a *feasible flow*, rather than a maximum flow, in a network where each node in the network has either a supply of flow or a demand for flow. Naturally, the supply must meet the demand. To model this we use a directed graph $G = (V, E)$ with capacities $c(u, v) \geq 0$, for each $(u, v) \in E$ and $c(u, v) = 0$ for $(u, v) \notin E$. In addition, for each vertex $v \in V$, there is a value $g(v)$ which represents the supply/demand for v. The quantity $g(v) > 0$ means that v has a supply, and $g(v) < 0$ means that v has a demand. Some nodes v may have $g(v) = 0$ which means that these nodes are only intermediaries for any flow, and do not have a supply or demand themselves. We require that
$$\sum_{v \in V} g(v) \geq 0,$$
that is, the total supply is at least the total demand. A *feasible flow* is a real valued function f on $V \times V$ with the properties:

   (a) $f(u, v) \leq c(u, v)$ for all $u, v \in V$,
   (b) $f(u, v) = -f(v, u)$ for all $u, v \in V$,
   (c) $\sum_{v \in V} f(u, v) = g(u)$ for all $u \in V$ with $g(u) \leq 0$.
   (d) $\sum_{v \in V} f(u, v) \leq g(u)$ for all $u \in V$ with $g(u) > 0$.

   Condition (c) states that all demands are met and condition (d) states that the supply at any node is never exceeded. Show how to reduce the feasible flow problem to the maximum flow problem. Show the reduction and argue effectively that it is correct.

2. In this problem we investigate a two-processor assignment of program modules. Assume that there are n program modules indexed $1, 2, \ldots, n$ and two processors A and B. If program module i is executed on one processor and program module j on the other then the time for interprocessor communication is $c_{ij} \geq 0$. We assume that $c_{ij} = c_{ji}$. The processors have different speeds so that program module i takes time A_i on processor A and B_i on processor B. The problem is to find an assignment of the program modules to processors that minimizes the total time. Reduce this problem to the max flow or the min cut problem. Argue briefly that your reduction is correct.

3. In this problem we will learn about the min cost max flow problem and examine a problem reducible to it. As usual we start with a directed graph $G = (V, E)$ with special vertices $s, t \in V$. We have the usual capacity function $c(u, v) \geq 0$ for $(u, v) \in E$ and $c(u, v) = 0$ for $(u, v) \notin E$. In addition, we have a *cost* $p(u, v) \geq 0$ for each $(u, v) \in E$. The quantity $p(u, v)$

is the cost per unit of flow on the edge (u, v). In the min cost max flow problem we seek a maximum flow f from s to t of minimum cost. That is, among all the maximum flows from s to t we seek the flow f that minimizes

$$\sum_{(u,v)\in E} p(u, v)f(u, v).$$

Min cost max flow has a polynomial time solution which is more complicated than the basic max flow problem. It has many applications among which is the following load balancing problem.

Suppose we are given a set of clients U and and set of servers V. For each client u there is a set $V_u \subseteq V$ of servers that are qualified to serve it. When a client is served it takes one unit of time. What we would like to compute is an assignment of the clients to the servers such that each client is assigned to a server that is qualified to serve it and that the total waiting time of all the clients is minimized. Suppose a server is assigned 4 clients, then they have to be served one at a time. The total waiting time for these 4 clients is 1 for the first, 2 for the second, 3 for the third, and 4 for the fourth, all totaling $1 + 2 + 3 + 4 = 10$.

(a) Formally define this load balancing problem as an optimization problem on bipartite graphs.

(b) Describe a reduction of this problem to the min cost max flow problem. Argue that your reduction is correct.