Network Optimization (Fall 2008) — Practice Final Exam

- This is an *open-book* exam.
- You are not supposed to discuss this exam with any one except me.

1. (20) Let $\boldsymbol{x}^*$ be an optimal solution of a minimum cost flow problem. Define the network G^0 as the subgraph of all arcs in $G(\boldsymbol{x}^*)$ with zero reduced cost. Show that the original minimum cost flow problem has an alternative optimal solution if and only if G^0 has a directed cycle.

2. (20) Consider the following DIVIDE AND CONQUER algorithm to solve the minimum spanning tree (MST) problem. Given graph $G = (N, A)$, partition N into two sets N_1 and N_2 such that their cardinalities differ by at most 1. Let $A_1 = \{(i, j) \in A \,|\, i \in N_1,\, j \in N_1\}$ and $A_2 = \{(i, j) \in A \,|\, i \in N_2,\, j \in N_2\}$. Recursively solve the MST problem on $G_1 = (N_1, A_1)$ and on $G_2 = (N_2, A_2)$. Let the MSTs for G_1 and G_2 be T_1 and T_2, respectively. Find the arc (k, l) such that $c_{kl} = \min\{c_{ij} \,|\, i \in N_1,\, j \in N_2\}$ (the cheapest arc in the cut), and set the MST of G as $T_1 \cup T_2 \cup \{(k, l)\}$.

   Will the final tree T returned by the algorithm be an MST for the original graph G? Justify your answer with a proof, or a counterexample, as appropriate.

3. (18) Let $G = (N, A)$ be a directed *acyclic* graph. For a subset of nodes $N_1 \subset N$, we define the out-arc list and in-arc list as follows:

   $$\mathcal{A}(N_1) = \{(i, j) \in A \,|\, i \in N_1, j \notin N_1\}, \quad \text{and} \quad \mathcal{AI}(N_1) = \{(j, i) \in A \,|\, i \in N_1, j \notin N_1\}.$$

   A subset of arcs $A_1 \subset A$ is called a *directed cut* if there is a subset of nodes N_1 with $\emptyset \neq N_1 \subset N$ such that $\mathcal{A}(N_1) = A_1$ and $\mathcal{AI}(N_1) = \emptyset$. Thus, each directed cut is also a cut. Similar to an s–t cut, a directed cut is called a directed s–t cut, if it separates the nodes s and t.

   (a) For nodes $s, t \in N$, prove that the maximum length *in terms of number of arcs* of an s–t path is equal to the minimum number of directed s–t cuts covering all arcs that are on at least one s–t path.

   (b) What will be the *weighted* version of the above result – when integer weights $c_{ij} \geq 0$ for each arc (i, j) are given, and we consider the maximum *sum of weights* of an s–t path ?

4. (22) A bipartite undirected graph $G = (N_1 \cup N_2, A)$ is called *k-regular* if $\text{degree}(i) = k \;\forall\, i \in N_1 \cup N_2$ (i.e., each vertex has the same degree). Use the max-flow min-cut theorem to prove that every k-regular bipartite graph has a perfect matching. Note: A *matching* is a subset of the arcs A such that each node involved is incident to *at most* one arc. A *perfect* matching in G is a matching that matches *all* nodes of G.

5. (20) Consider a round-robin chess tournament with n players with each player playing every other player exactly once. A win scores 1 for the winner and 0 for the loser, while a draw scores 1/2 for each player. We are given a set of final scores $(s_1, \ldots, s_n)$ for the players with $0 \leq s_i \leq n - 1$. We want to check whether these scores are feasible (for example, in a three-player tournament, a set of final scores of $(2, 1/2, 2)$ is impossible). Model this problem as a network flow problem.