CMPSCI 611: Advanced Algorithms
Micah Adler

Problem Set 3

Out: October 31, 2001
Due: November 7, 2001

1. Consider a hockey league with n teams $T_1 \ldots T_n$. At a certain point in the season, we want to determine if team T_1 has been *mathematically eliminated*. In other words, if, no matter what happens in the remaining games, some other team will end up with more wins than team T_1. In this league, there are no ties.

   The input will consist of a set of integers $w_1 \ldots w_n$, where w_i is the number of wins that team T_i has thus far in the season, as well as c_{ij}, for $1 \leq i < j \leq n$, which represents the number of remaining games between teams i and j.

   (a) Show that the problem of deciding whether team T_1 is mathematically eliminated can be formulated as a maximum flow problem.

   Hint: first describe why you can assume that T_1 wins the rest of its games. Then focus on the values $\ell_2 \ldots \ell_n$, where ℓ_i is the number of losses that team T_i must have to not eliminate team T_1.

   (b) By applying the max-flow min-cut theorem to your network from part (a) provide and prove correct a simple necessary and sufficient condition for team T_1 being mathematically eliminated. This condition should involve the existence of a subset of teams with a certain property.

2. [CLR] Problem 27-2 (page 626 of first edition) ***Minimum path cover***.

3. Given an undirected graph $G = (V, E)$, and vertices s and t, let's say we want to find a minimum s-t cut. We here consider adapting the technique used for Karger's minimum cut algorithm to this problem. In the resulting s-t cut algorithm, as in the general cut algorithm, vertices are merged by contracting edges. Vertices s and t may be merged with other vertices; we call the vertex containing s the s-vertex, and the vertex containing t the t-vertex. Since we want to find the minimum s-t cut, if we ever choose an edge that would merge the s-vertex with the t-vertex, we do not contract that edge. Instead, the current graph is not changed, and we choose a new random edge. At the end of the algorithm, we are left with an s-vertex and a t-vertex, and we return the set of edges connecting these two vertices.

   Show that there are graphs in which the probability that this algorithm finds a minimum s-t cut is exponentially small in $|V|$. What does this imply about how useful this algorithm is?

4. Say we are given two lists L_1 and L_2, each containing n integers, and we want to determine if the two lists have the property that each integer occurs the same number of times in both sets. This problem could be solved by sorting the two lists in $O(n \log n)$ time, and then comparing the two sorted lists.

   Describe a randomized technique for solving this problem based on verifying a polynomial identity. Demonstrate a bound on the probability of the resulting algorithm returning "yes" when the correct answer is "no" and a bound on the probability of returning "no" when the correct answer is "yes". Describe how to make these probabilities smaller than ϵ, for any given ϵ. Describe the running time of your algorithm, assuming that pairwise arithmetic operations on arbitrarily large integers can be performed in constant time.