CS466/666, Fall 2009: Assignment 5

Out: November 16, Due: December 2, 5pm

1. **Max-ℓ-Cut.** Given an undirected graph $G = (V, E)$ and an integer ℓ, the *Max-ℓ-cut* problem asks to find a partition of V into sets $S_1, \ldots, S_\ell$ such that the number of edges in the cut (i.e., connecting vertices in two different sets) is maximized. cut (i.e., connect vertices in two different sets) is maximized.

   (a) A simple randomized algorithm would assign each vertex to one of $S_1, \ldots, S_\ell$ with equal probability. What is the expected number of edges in the ℓ-Cut achieved with this algorithm?

   (b) Explain how to de-randomize the algorithm of part (a) with the method of conditional expectation. Then give a simple interpretation of the resulting algorithm as a greedy-algorithm.

2. **NAE-6SAT:** NAE-6SAT is the same as NAE-3SAT, except that every clause has exactly 6 distinct literals. NAE-6SAT is still NP-hard (you need not prove this.)

   But it is polynomial in a special case. Assume that it is known that there exists an assignment R to the variables such that every clause has exactly 3 literals that are TRUE. (You do not know what R is, only that it exists.) Show that then there exists a Las Vegas algorithm that finds a solution to this instance of NAE-6SAT in polynomial expected time.

   (Hint: Imitate the randomized algorithm for 2-SAT. Note that you need not find R; any solution to NAE-6SAT is good enough.)

3. **Nonograms:** The following is a very specialized case of a nonogram. Assume you are given $h_1, \ldots, h_m$ and $v_1, \ldots, v_n$ with $\sum_{i=1}^m h_i = \sum_{j=1}^n v_j$. You are also given two functions $\alpha : \{1, \ldots, m\} \to \{1, \ldots, n\}$ and $\beta : \{1, \ldots, n\} \to \{1, \ldots, m\}$.

   (a) You want to find a 0/1-matrix $(x_{i,j})$, where $i = 1, \ldots, m$ and $j = 1, \ldots, n$ that satisfies the following:

   - $\sum_{j=1}^n x_{i,j} = h_i$ for all $i = 1, \ldots, m$.
   - $\sum_{i=1}^m x_{i,j} = v_j$ for all $j = 1, \ldots, n$.
   - For any $i = 1, \ldots, m$, the 1s in row i are consecutive.
   - For any $j = 1, \ldots, n$, the 1s in column j are consecutive.
   - For any $i = 1, \ldots, m$, $x_{i,\alpha(i)} = 1$. $(*)$
   - For any $j = 1, \ldots, n$, $x_{\beta(j),j} = 1$. $(**)$

   Show how to find such a matrix in $O(mn)$ time. Hint: 2SAT.

   (b) Show how to do this if you *don't* have function $\beta(.)$ (and condition $(**)$ is dropped.) The goal is still $O(mn)$ run-time, but slower methods will receive lots of partial credit.

   (c) (Bonus) Show how to this is you have neither $\alpha(.)$ nor $\beta(.)$ (and conditions $(*)$ and $(**)$ are dropped), but the set of cells with $x_{i,j} = 1$ is required to be connected. The run-time should still be polynomial, but not particularly fast.

4. **k-Vertex-Cover:** A different way to phrase the Vertex Cover problem is to ask for the maximum number of edges that can be covered with a set of k vertices. Thus, given a graph and k, we want to find a set $C \subset V$ of k vertices such that as many edges as possible have an endpoint in C.

   Give a randomized algorithm that finds a set $C \subset V$ such that the expected size of C is at most k and the expected number of covered edges is at least $\frac{3}{4}OPT$, where OPT is the maximum number of edges that can be covered with k vertices. (Hint: Randomized rounding. You will need to formulate a suitable IP first.)