Algorithms, Winter 2010-11, Homework 8
due Friday 2/18/11, 4:00pm

Problem 1

Given is an $n \times m$ array $Chessboard$ containing only 0's (empty squares) and 1's (occupied positions). Moreover, there is an unlimited number of dominos, i.e. 1×2 rectangle pieces. Your task is to decide if it is possible to cover all of the empty squares on the chessboard by nonoverlapping dominos (the dominos cannot cover any of the occupied positions and they cannot "stick out" of the chessboard). Your algorithm should run in time $O((mn)^3)$.

Include a short paragraph describing your algorithm, briefly argue your algorithm's correctness, and explain its running time.

Problem 2

Recall that the Edmonds-Karp algorithm refines the idea of Ford and Fulkerson in the following way: in every iteration, the algorithm chooses the augmenting path that uses the smallest number of edges (if there are more such paths, it chooses one arbitrarily). Find a graph for which in some iteration the Edmonds-Karp algorithm has to choose a path that uses a backward edge. Run the algorithm on your graph – more precisely, for every iteration draw the residual graph and show the augmenting path taken by the algorithm as well as the flow after adding the augmenting path.

Problem 3

The Traveling Salesman Problem (TSP) is defined as follows: given is a complete weighted undirected graph on n vertices (i.e., there is an edge between every pair of vertices) and a number $k > 0$, does there exist a cycle going through every vertex exactly once with total weight at most k? (The weight of a cycle is the sum of the weights of the edges forming the cycle.)

The Hamiltonian Cycle (HC) problem is defined as follows: given is an undirected graph, does there exist a cycle that goes through every vertex exactly once?

Show that HC is polynomially-reducible to TSP, i.e., HC $\leq_P$ TSP. In other words, assume that we have a black box that solves TSP (the input to the black box is n, the weights of all the edges, and the number k). We need to:

(a) Let G be an input of HC. Transform it into an input for the black box.

(b) After the black box produces an answer (YES or NO), transform it into the answer to HC with input G.

Your solution should describe both steps (a) and (b) and argue why the construction works.

Note: both TSP and HC problems are known to be NP-complete. This means that we do not know if they can be solved in polynomial time but most people think that no polynomial-time algorithm exists. Nevertheless, we can pretend to have a black box for TSP and see if such black box would help solving HC (and other problems).