Algorithms, Winter 2010-11, Homework 5
due Friday 1/28/11, 4:00pm

Problem 1

Given is a convex polygon with n vertices $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ (the vertices are listed in a clockwise order). Recall that a triangulation of a convex polygon is a set of $n-3$ non-intersecting edges, where each edge connects two non-consecutive vertices (the overall picture consists of $n-2$ triangles that together form the original polygon). We will define the *length* of a triangulation as the sum of the lengths of these $n-3$ edges. Give an $O(n^3)$ algorithm that finds the minimum possible length of a triangulation of the given polygon.

Hint: Use a 2D dynamic programming array. As before, the heart of the solution that consists of three parts (the verbal description, the mathematical formula that computes the described value, and the return value) accounts for 50% of the score.

Problem 2

Give an algorithm that finds the number of connected components of a given graph G. (Two vertices are in the same connected component of G if there exists a path between these vertices.) Your algorithm should run in linear time, i.e., $O(n+m)$, where n is the number of vertices and m is the number of edges. Describe your algorithm in words and pseudocode (as well as submit a Java/C/C++ implementation) and reason its running time.

Problem 3

Given is an $m \times n$ array of zeros and ones, plus exactly one number 2 and exactly one number 3. The array represents a maze, where zeros correspond to empty positions and ones correspond to walls. Numbers 2 correspond to a person and number 3 corresponds to the person's house. The person can move from one empty position to another adjacent position (every position has at most four adjacent positions – east, west, north and south). Give an $O(mn)$ algorithm that outputs the length of the shortest path from position 2 to position 3. Describe your algorithm in words and pseudocode (as well as submit a Java/C/C++ implementation) and reason its running time.