ASSIGNMENT 2

ACKNOWLEDGE YOUR SOURCES.

1. [10 marks] Give a linear-time algorithm to find the convex hull of the union of two convex polygons in the plane. Your algorithm should handle any pair of convex polygons, disjoint, nested, etc.

   **Hint.** Think about which of the convex hull algorithms we covered in class can be used for this purpose.

2. [10 marks] Steinitz's theorem from 1922 is a fundamental result about the combinatorics of 3-dimensional convex polyhedra. It says that a graph is the graph of a 3D convex polyhedron if and only if the graph is planar and 3-connected. Note that a triangle is a degenerate polyhedron but it's really better to start with the complete graph on 4 vertices.

   The easy direction of the proof is that a convex polyhedron yields a planar 3-connected graph. (Think about it.)

   (a) [7 marks] Prove the hard direction for the special case of a graph that is an “Apollonian network” or “planar 3-tree” defined as follows: start with a triangle and repeatedly choose a triangular face $abc$ and subdivide it into 3 triangles by adding one new vertex $x$ with edges to $a$, $b$, and $c$. Please draw some examples, or see the wikipedia definition of “Apollonian network”.

   Your proof should provide an efficient algorithm to construct the convex polyhedron for any planar 3-tree. (Note: Looking on the internet may lead you astray into the realm of “lifting algorithms”—you don’t need anything so complicated. I recommend trying the problem yourself before searching.)

   (b) [3 marks] What model of computation does your algorithm assume? Do you need a real RAM?

   **Hint 1.** You may use the following fact. Define the size of a rational number $\frac{a}{b}$ to be the number of bits in $a$ and $b$. Define the size of a linear equation to be the sum of the sizes of all coefficients. Let $A$ be a system of linear equations. Then there is a polynomial time algorithm (counting bit operations) to find a solution to $A$ if one exists. In particular, if there is a solution, there is one of polynomial size. The algorithm is a careful version of Gaussian elimination, and Schrijver’s book, “Theory of Linear and Integer Programming,” is a good source.

   **Hint 2.** This fact may be slightly less useful than it first appears.