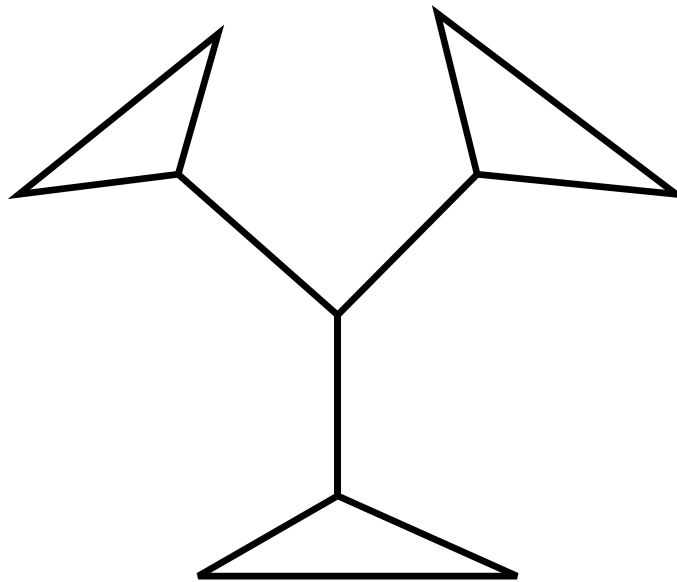


1. *20 points.* **3-dimensional convex hulls.** Suppose you are given two disjoint convex polygons P and Q in \mathbb{R}^3 . Recall that the *sleeve* of P and Q is defined to be the set of triangular faces of the convex hull of $P \cup Q$ that do not already appear on either P or Q . Define the *trace* of the sleeve on P to be the intersection of the sleeve with P , which defines a piece-wise linear curve. In class I showed that the trace of the sleeve on P need not be simple, and can be topologically equivalent to a “bar bell.” Construct a physical model (out of paper, cardboard, coat hangers, clay, ...) to show that the trace of the sleeve on P can be topologically equivalent to a 3-armed “windmill”, as drawn below:



Note that two such convex polygons define a counter-example to the stopping criterion of just returning to a starting vertex—you must stop by returning to the starting edge.

2. Let P be a convex polyhedron P with n vertices, stored in your favorite data structure.
- Describe how to compute the *shadow* of P , given a light source at infinity in direction v . (The *shadow* is the polygon that is the projection of P onto a plane perpendicular to v .)
 - What is the geometric dual of the shadow?
3. Let A and B be two disjoint n -sized sets of points in the plane.
- Develop an $O(n \log n)$ -time algorithm for determining for each point p in A (resp., B) the nearest neighbor point of p in B (resp., A).

(b) The *Hausdorff distance* between A and B is

$$\max_{a \in A} \min_{b \in B} d(a, b),$$

where $d(a, b)$ is the Euclidean distance from a to b . This metric is used to measure similarity—it is the furthest distance an A -point can be found from its closest B -neighbor. Outline how to compute the Hausdorff distance in $O(n \log n)$ time.

4. Given a set of n lines $\{\ell_1, \ell_2, \dots, \ell_n\}$ in the plane, each having a weight $w(\ell_i)$, and a large set of m points P —assume that m is comparable to n^2 . For each point $p \in P$, compute the weight of the lines beneath p (i.e., the weight of those intersected by a vertical ray emanating downward from p). What is the running time of your method?
5. Given n points, develop an algorithm for determining if any three are colinear. What is the running time of your method?