Stephen Ranger
Computational Geometry
Homework 1
Due Date: 2015-02-22

Given a set S of disjoint triangles, finding a set of n-1 lines to connect each in O(n log n) time could be defined as follows:

---
for minX <= x <= maxX
    check intersections with triangles
        - if intersection occurs, mark corner of triangle
            + follow connecting edges, marking other two points until done

        - if parallel with line, mark corners at end of edge
            + follow connecting edges, marking final point

        - if a current triangle is being tracked and a new triangle begins, draw line between them
            + if multiple triangles being tracked, use the triangle closest to new triangle

        - if a new triangle is found without a triangle currently tracking, draw line between it and previously seen triangle

        - mark newly found triangle as newest

For my data structures, I would store a list of currently tracking triangles as well as a list of output lines. As I sweep across, I would insert lines between triangles as they occur.

The sweeping will fall into the following categories:
1. Line intersects single vertex
    a. Mark as start of new triangle
2. Line intersects parallel line
    a. Mark as edge of new triangle
3. Line intersects existing line
    a. determine which triangle it belongs (slope and intercept of previous lines can be used)

Of the following, during 1 and 2, we will be finding a new triangle. In these cases, the new triangle will be connected to the most recently found triangle; if two exist, use the one closest. All three can occur concurrently.