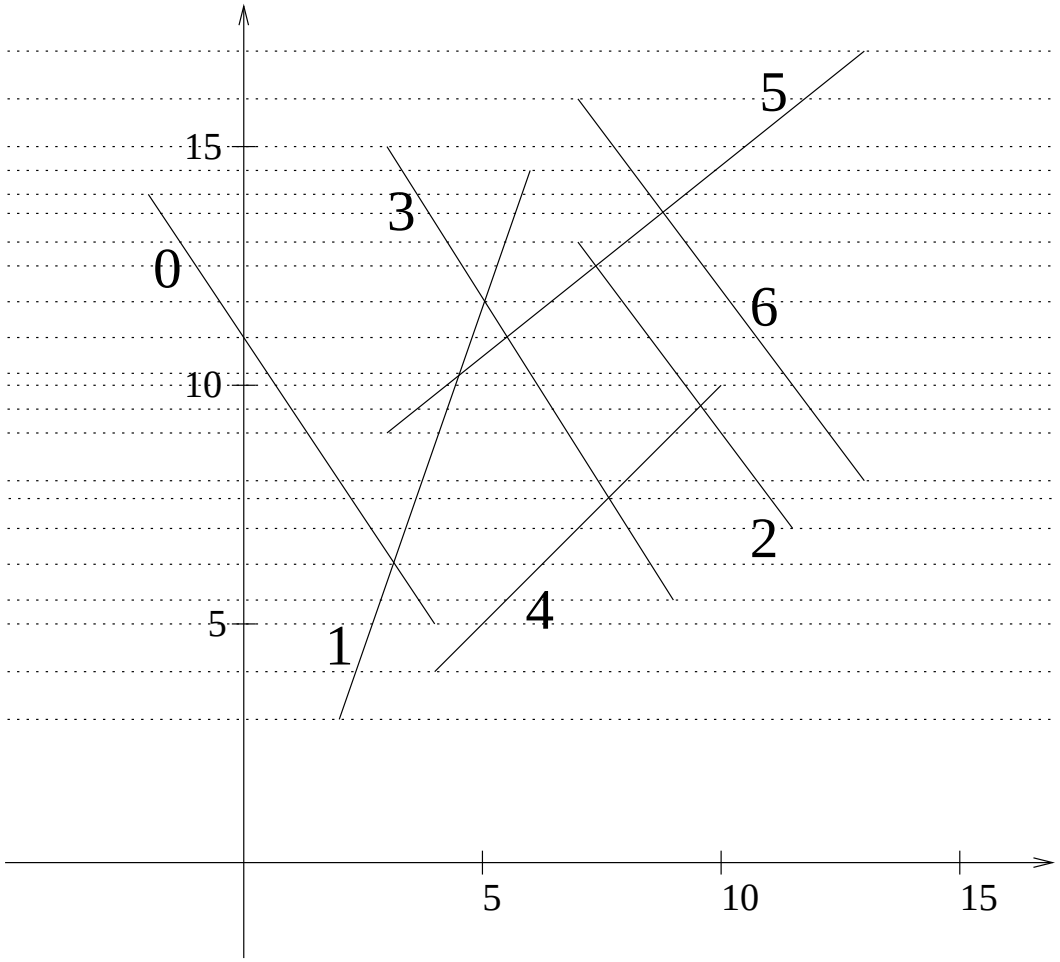
COMPUTATIONAL GEOMETRY
Homework Set # 8

Due by 4pm (to my office or mailbox), Firday, May 9, 2008.

Recommended Reading: O'Rourke, Chapter 7 (you may omit sections 7.3, 7.5, 7.6, 7.8, 7.9).

In all of the exercises, be sure to give at least a brief explanation or justification for each claim that you make.

(1). [25 points] Consider the set S of 7 line segments given by $S = \{s_0, s_1, \ldots, s_6\} =$ { ((-2,14),(4,5)), ((6,14.5),(2,3)), ((7,13),(11.5,7)), ((3,15),(9,5.5)), ((10,10),(4,4)), ((13,17),(3,9)), ((7,16),(13,8))}, where each segment $s_i = (a_i, b_i)$ has upper endpoint a_i and lower endpoint b_i. I draw horizontal lines through the endpoints and the intersection points in the figure, to help you order the events.



Apply the Bentley-Ottmann sweepline algorithm to S. Give the event queue Q and the sweep status $\mathcal{L}$ just after each event. (Use the notation as in the text that x_{ij} denotes the point (if any) at the intersection of segment s_i and segment s_j.)

Show the chart, just as in the handout given in class. (You need not compute the actual intersection points x_{ij}.)

You may use either the "usual" Bentley-Ottmann sweep or the "modified" version (which removes from the event queue those crossing events that no longer correspond to segments that are adjacent in the sweep line status (SLS)). But please *state* which method you are using. (My solutions will show both, as on the handout, using the notation that an entry "$[x_{ij}]$" means that the event point x_{ij} would appear in the usual algorithm but not in the modified algorithm. You should understand both versions.)

(2). [20 points] O'Rourke, problem 4, section 7.11.5, page 293.

(3). [20 points] O'Rourke, problem 5, section 7.11.5, page 293.

(4). [35 points] Build the Kirkpatrick point location hierarchy for the triangulation shown below. At each step, when you identify an independent set, apply Algorithm 7.4 on page 277, breaking ties when you select a node in favor of the lowest numbered vertex. When you retriangulate a hole, use the simple ear-clipping algorithm (`Triangulate`, page 39), starting at the bottommost vertex of the hole (as "$v0$" in `Triangulate`, the first one tested for earity), and proceeding counterclockwise. (Ties (if any) for bottommost should be broken by picking the rightmost among the bottommost vertices.)

(a). List the independent sets corresponding to each stage of the algorithm. Also, for each stage, draw the corresponding triangulation.

(b). Draw the final hierarchy as a DAG, with each node of the hierarchy labeled by the triangle to which it corresponds. (When you label a node, please list the triangle as a triple with the vertex indices in order; e.g., the triangle with vertices "1", "8" and "9" should be written as "189" (not as "819" or "918", etc).)

(c). Highlight in the final hierarchy those nodes that are explored when point location is performed for point p, as shown in the figure.