Homework 3 Solution

Elisha Sacks

Problem 1

(a) Prove that two convex polygons with a total of n vertices have at most n intersection points.

The intersection polygon p has at most n edges, since every input edge contributes at most one edge by convexity. Hence, p has at most n vertices. Every intersection point is a vertex of p.

(b) How does this improve the asymptotic running time of the sweep algorithm?

The running time is $O(n \log n)$ because the output size is $O(n)$.

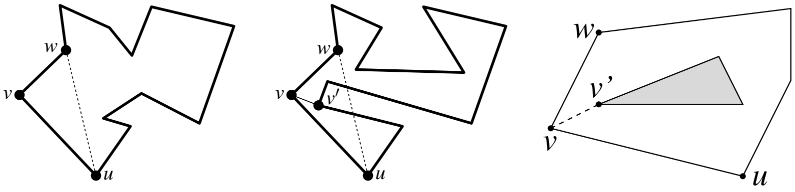
Problem 1

(c) Describe a modified sweep algorithm that computes the intersection points in $O(n)$ time.

Initialize the event queue with the edges incident on the leftmost input vertex. When handling a right endpoint event, enqueue the edges with this left endpoint.

Creating the initial edges and initializing the queue take $O(n)$ time. There are $O(n)$ events: $2n$ endpoints and at most n swaps. Event handling takes constant time because the sweep and the queue have bounded size. The sweep contains at most four edges because a line intersects a convex polygon at most twice. These edges have at most two intersections. Hence, the queue contains at most four endpoint events and at most two swap events.

Problem 2



(a) Explain why the proof that every polygon has a triangulation still works for polygonal regions with inner boundaries or fix it if necessary.

The induction is on the number of vertices n and the number of boundaries m with base case $n = 3$ and $m = 1$. Define u, v, w, and v' as before. If uw is a diagonal or v' is on the outer boundary, the diagonal splits the polygon into two polygons with fewer than n vertices. Otherwise, the diagonal reduces m by one.

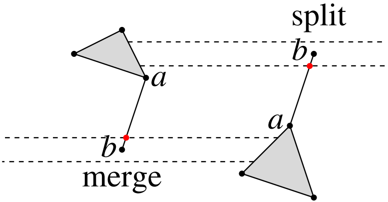
The number of triangles is no longer $n - 2$.

Problem 2

(b) The incoming and outgoing edges of vertex v_i are no longer e_{i-1} and e_i. Explain how to find them using the doubly linked list representation.

Each boundary loop of the face is represented by an edge e. Form a list of the edges on the loop by following the next links until returning to e. Consecutive edges ab and bc are the incoming and outgoing edges of b.

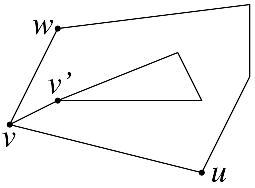
Problem 2



(c) Explain how dangling edges are handled.

An edge ab is dangling if its twin equals its next. It is a split if $a_y < b_y$ and is a merge if $a_y > b_y$. These splits and merges are handled as before.

Problem 2



What about general subdivisions?

Problem 3

Explain why the greedy triangulation algorithm treats the last point specially.

The last point is on both chains. It can be assigned to one chain based on the next to the last point. Even so, the standard logic can create a diagonal that duplicates a polygon edge.