COM S 418 Assignment 2

Eric Hedgren

Question 2.5

Which of the following equalities are always true?

1. **Twin(Twin($\vec{e}$)) = $\vec{e}$**
   Always true. twin of a twin is itself, thus it will return to the same edge.
2. **Next(Prev($\vec{e}$)) = $\vec{e}$**
   Always true. Prev counter acts the next, thus it will return back to itself.
3. **Twin(Prev(Twin($\vec{e}$))) = Next($\vec{e}$)**
   Not always true. The prev of the twin does have to be itself.
4. **IncidentFace($\vec{e}$) = IncidentFace(Next($\vec{e}$))**
   Always true. the next pointer keeps us on an edge for the same face

Question 2.9

Suppose that a doubly-connected edge list of a connected subdivision is given. Give pseudocode for an algorithm that lists all faces with vertices that appear on the outer boundary.

First, find the leftmost vertex. Next, follow the edge with the largest slope, since it will have to be on the outer boundary, continue using the Next pointers from that edge until we return to the starting vertex. At each vertex that we have visited, we will find all the faces that it touches.

Question 2.12

Let S be a set of n triangles in the plane. The boundaries of the triangles are disjoint, but it is possible that a triangle lies completely inside another triangle. Let P be a set of n points in the plane. Give an O(n log n) algorithm that reports each point in P lying outside all triangles.

To accomplish this we can use the sweep line algorithm. Sort triangles in a set S based on their y-coordinates, high-to-low. Initialize a BST as a status structure to maintain the active set of triangles intersecting the sweep line.

Sweep a horizontal line from the highest y-coordinate to the lowest of the points in P. At each event update the status structure accordingly:

1. for point P check its position relative to the active set of triangles using the status structure. Check if it lies outside all triangles in the status structure. If so report it.
2. for top vertex of a triangle, insert the triangle into the status structure
3. for the bottom vertex of a triangle, remove it from the status structure

The final result ends with a list of all the points in P that lie outside all the triangles in S. Because of using the sweep line algorithm the run time of it is O(n log n).

Question 2.14

Let S be a set of n disjoint line segments in the plane, and let p be a point not on any of the line segments of S. We wish to determine all line segments of S that p can see, that is, all line segments of S that contain some point q so that the open segment pq doesn't intersect any p not visible line segment of S. Give an O(n log n) time algorithm for this problem that uses a rotating half-line with its endpoint at p.

To accomplish this we can modify the line sweep algorithm. Accumulate the endpoints into an array the sort by their polar angles with respect to p. Instead of using a vertical line and sweeping from left-to-right we are going to have a line rotate around the point p.

At intersection points, when the rotating line intersects with an endpoint, check for visibility in relation to p, and update the set containing line segments. To check for visibility examine the angles formed by the rotating line and line segments in the active set whether or not a line segment is intersecting pq. From sorting the points and the sweeping steps, this algorithm is O(n log n).

Question 3.2

A rectilinear polygon is a simple polygon of which all edges are horizontal or vertical. Let P be a rectilinear polygon with n vertices. Give an example to show that ⌊n/4⌋ cameras are sometimes necessary to guard it.

A simple square is an example of this. Since there are 4 vertices in a square, then there has to be $\lfloor \frac{4}{n} \rfloor$ = 1 camera to cover this area. Any less than 1 camera and it will not be covered.