Point Location (chapter 6)

Elisha Sacks
Point Location

Find the face of a subdivision that contains a query point.
Form slabs with verticals through the subdivision vertices.

Sort the slabs along the $x$ axis.

Sort the edges of each slab along the $y$ axis.

Locate a point via two binary searches.
Complexity

- Good point location time: $O(\log n)$.
- Bad space complexity: $O(n^2)$.
- Bad preprocessing time: $O(n^2 \log n)$. 
Trapezoidal Map

Overlay the subdivision with a box $R$ and subdivide the bounded faces with verticals from each vertex to the edges above and below.
The faces of the trapezoidal map are trapezoids and triangles.

The top and bottom sides are segments of overlay edges.

The left and right sides are verticals through overlay vertices.

The left or right side of a triangle is a subdivision vertex.

The straightforward proof appears in the textbook.

We assume that subdivision vertex $x$ coordinates are distinct.

This assumption will be removed later.
A trapezoid $\Delta$ is represented by its $\text{top}(\Delta)$ and $\text{bottom}(\Delta)$ edges and by its $\text{leftp}(\Delta)$ and $\text{rightp}(\Delta)$ vertices.

The figure shows the cases for $\text{leftp}(\Delta)$.

The cases for $\text{rightp}(\Delta)$ are analogous.

The non-subdivision vertices are represented implicitly.
Lemma 6.2 A trapezoidal map of a subdivision with $n$ edges has at most $6n + 4$ vertices and $3n + 1$ trapezoids.

Proof

Vertices: 4 from $R$, $2n$ from the subdivision, and $2 \times 2n$ from the vertical sides.

Trapezoids: Every trapezoid has a $leftp$. The bottom of $R$ is the $leftp$ of one trapezoid, a subdivision edge defines at most two $leftp$ with its left endpoint and at most one $leftp$ with its right endpoint.
Lemma 6.2 A trapezoidal map of a subdivision with \( n \) edges has at most \( 6n + 4 \) vertices and \( 3n + 1 \) trapezoids.

Proof

Vertices: 4 from \( R \), \( 2n \) from the subdivision, and \( 2 \times 2n \) from the vertical sides.

Trapezoids: Every trapezoid has a \( leftp \). The bottom of \( R \) is the \( leftp \) of one trapezoid, a subdivision edge defines at most two \( leftp \) with its left endpoint and at most one \( leftp \) with its right endpoint.

Why doesn’t \( leftp(\Delta) \) define three trapezoids in (a)?
Distinct \( x \) coordinates imply that a trapezoid has at most two left neighbors and at most two right neighbors (i).

The left neighbors are encoded by top, bottom, and \( \text{leftp} \).

The right neighbors are encoded by top, bottom, and \( \text{rightp} \).

Each trapezoid stores pointers to its neighbors.

Duplicate \( x \) coordinates allow any number of neighbors (ii).
Find the trapezoid containing a point \( a \) by traversing a graph.

- A point node \( u \) (white) branches left if \( a_x < u_x \).
- An edge node \( s \) (grey) branches left if \( a \) is above \( s \).
- The leafs point to trapezoids.
Incremental Construction of the Trapezoidal Map

1. Create a search graph for the bounding box \( R \).
2. Process the edges \( s_i \) of the subdivision in random order.
   2.1 Find the trapezoids that \( s_i \) intersects.
   2.2 Update the trapezoids.
   2.3 Update the search graph.
Finding the Trapezoids

1. Find the trapezoid $\Delta_0$ that contains the left endpoint using the search graph.

2. Move right: $\Delta_i$ is the upper/lower right neighbor of $\Delta_{i-1}$ if $s_i$ is above/below rightp($\Delta_{i-1}$).

3. Stop at the trapezoid $\Delta_k$ that contains the right endpoint.
Suppose $pr$ is inserted when $pq$ is in the graph.

- The $p_x$ test and the $pq$ test are identities.
- These cases are handled by secondary tests.
  - The $p_x$ test branches right.
  - The $pq$ test branches right when $LT(q, p, r) > 0$. 
Update: One Trapezoid
Update: General Case

\[
\Delta_0 \rightarrow \Delta_1 \rightarrow \Delta_2 \rightarrow \Delta_3
\]

\[
\mathcal{D}(S_{i-1}) \rightarrow \mathcal{D}(S_i)
\]

\[
\mathcal{T}(S_{i-1}) \rightarrow \mathcal{T}(S_i)
\]
Degeneracy

- A query point on an edge reports the edge.
- A query point equal to a vertex reports the vertex.
- Equal $x$ coordinates can be eliminated by input perturbation or by symbolic perturbation.
Symbolic Perturbation

- Shear the vertices: \((x, y) \rightarrow (x + \epsilon y, y)\).
- The \(x\) order is preserved for small enough positive \(\epsilon\).
- Predicates are evaluated *without* computing \(\epsilon\) or shearing.
  - The \(x\) order is replaced by lexicographic order.
  - The point/edge predicate is unchanged.
Symbolic Perturbation

- Shear the vertices: \((x, y) \rightarrow (x + \epsilon y, y)\).
- The \(x\) order is preserved for small enough positive \(\epsilon\).
- Predicates are evaluated \textit{without} computing \(\epsilon\) or shearing.
  - The \(x\) order is replaced by lexicographic order.
  - The point/edge predicate is unchanged.
- Does this strategy apply to algorithms that construct vertices?
Theorem 6.3 The trapezoidal map of $n$ segments has $O(\log n)$ query time, $O(n)$ space complexity, and $O(n \log n)$ construction time in randomized expectation.

Proof We will prove each bound in turn.
Expected Query Time

- The segments are inserted in random order $s_1, \ldots, s_n$.
- The first $i$ segments are $S_i = \{s_1, \ldots, s_i\}$.
- Let $P_i$ be the probability that inserting $s_i$ creates nodes on the path through the search graph to the trapezoid that contains $q$.
- Let $\Delta_q(S_i)$ denote the trapezoid of $T(S_i)$ that contains $q$.
- Key fact: $P_i = Pr[\Delta_q(S_i) \neq \Delta_q(S_{i-1})]$.
- If $s_i$ is removed, $\Delta_q(S_i)$ vanishes with probability $\leq 4/i$.
  - $s_i$ equals its top or bottom.
  - $s_i$ is the only segment incident on its leftp or rightp.
- Let $X_i$ nodes on the path to $q$ be created when $s_i$ is inserted.
- Case analysis shows that $X_i \leq 3$, so $E[X_i] \leq 3P_i$.
- The query time for $q$ is linear in the sum $X = X_1 + \cdots + X_n$.
- Using linearity of expectation

$$E[X] = \sum_{i=1}^{n} E[X_i] \leq \sum_{i=1}^{n} 3P_i \leq \sum_{i=1}^{n} \frac{12}{i} = 12H_n < 12(\log n + 1)$$
Expected Space Complexity

- The space complexity is proportional to the number of nodes.
- Let $k_i$ be the number of trapezoids created by inserting $s_i$.
- The graph grows by $k_i$ leafs and $k_i - 1$ internal nodes.
- The number of nodes is bounded by $2(k_1 + \cdots + k_n)$.
- Define $\delta(t, s)$ to equal 1 if $t$ vanishes from $T(S_i)$ when $s$ is removed and to equal 0 otherwise.
- Average $k_i$ over the $i$ choices of $s_i$ within $S_i$.

$$E[k_i] = \frac{1}{i} \sum_{s \in S_i} \sum_{t \in T(S_i)} \delta(t, s) = \frac{1}{i} \sum_{t \in T(S_i)} \sum_{s \in S_i} \delta(t, s)$$

$$\leq \frac{1}{i} \sum_{t \in T(S_i)} 4 = \frac{4|T(S_i)|}{i} \leq \frac{4(3i + 1)}{i} < 13$$

- The space complexity is $2 \sum_{i=1}^{n} E[k_i] = O(n)$. 
Expected Construction Time

- Time for $s_i$ is $O(\log i)$ lookup of left endpoint plus $E[k_i] = 1$.
- Construction time is $\sum_{i=1}^{n}(\log i + 1) = n \log n$. 