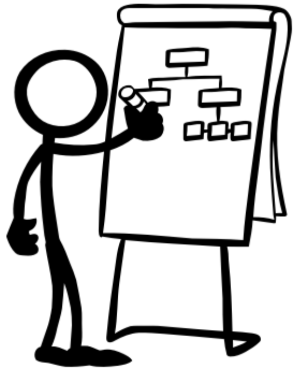
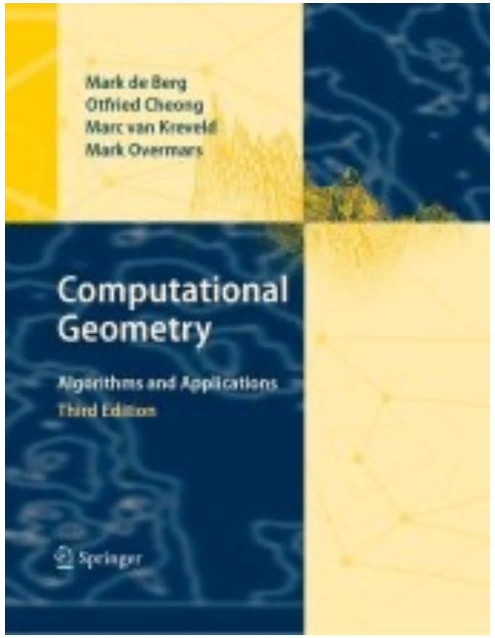
Point Location

Computational Geometry – Recitation 6



Agenda

We will solve various problems from the Point Location chapter in the course book.



Trapezoidal Map: Warm Up

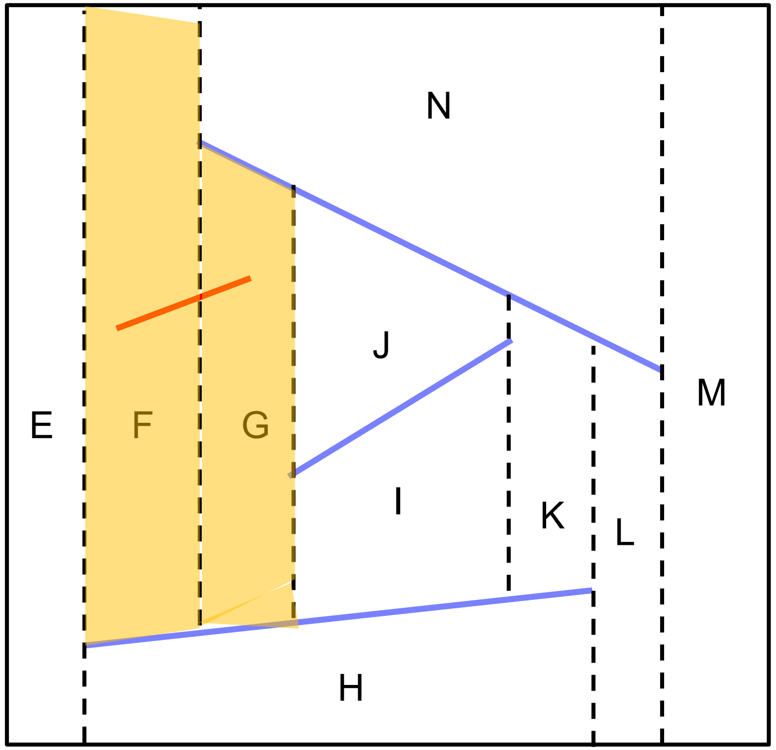
- Let S be a set of non-intersecting segments, and $T(S)$ be its trapezoidal map.
- Let s be a new segment not crossing any of the segments in S.

Prove that:

a trapezoid t in $T(S)$ is also a trapezoid in $T(S \cup s)$ $\Leftrightarrow$ s does not intersect t.

Trapezoidal Map: Warm Up

a trapezoid t in $T(S)$ is also a trapezoid in $T(S \cup s)$ $\Longleftrightarrow$ s does not intersect t.



Trapezoidal Map: Warm Up

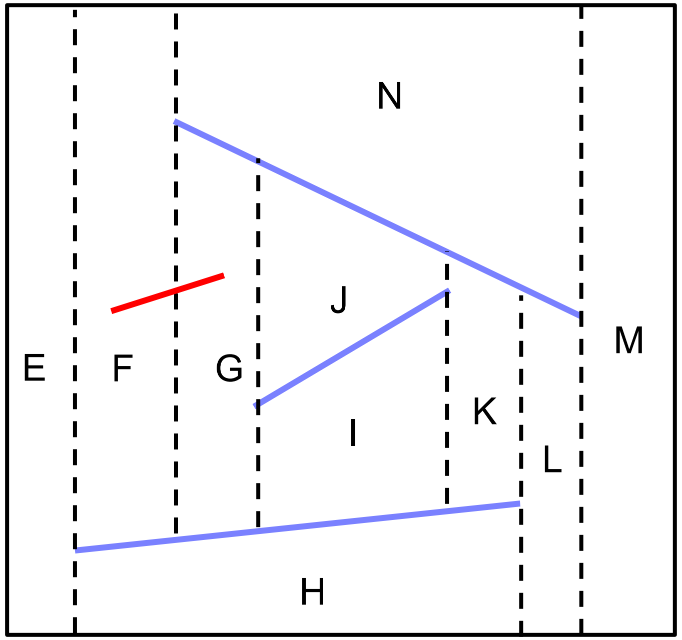
$\Leftarrow$ **If s does not intersect $\boldsymbol{t}$ then $\boldsymbol{t}$ belongs to $\boldsymbol{T(S \cup s)}$.**

- The algorithm we have seen to compute a trapezoidal map add the segments one by one.
- If we will add s as the last segment, then it clearly won't affect t (which is already part of the map) QED.

Trapezoidal Map: Warm Up

⇒ **If t belongs to $T(S \cup s)$ then s does not intersect t.**

- Assume by contradiction that *s* intersects *t,* then clearly *t* is splitted.

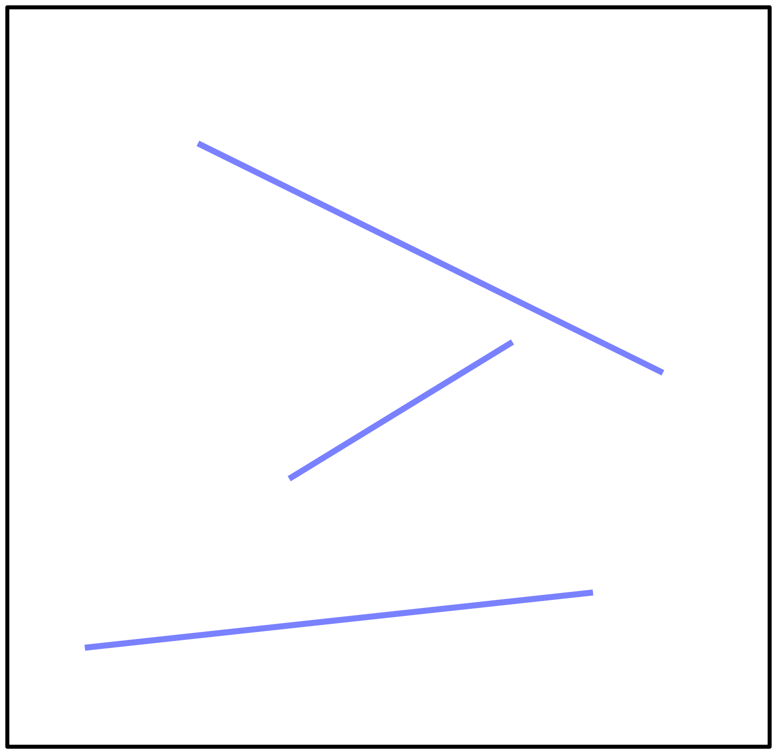


Q2 - Trapezoidal Map Computation

Design a deterministic algorithm to compute the trapezoidal map of a set of segments.

- No need to compute the search DAG, just the subdivision (as a DCEL for example).

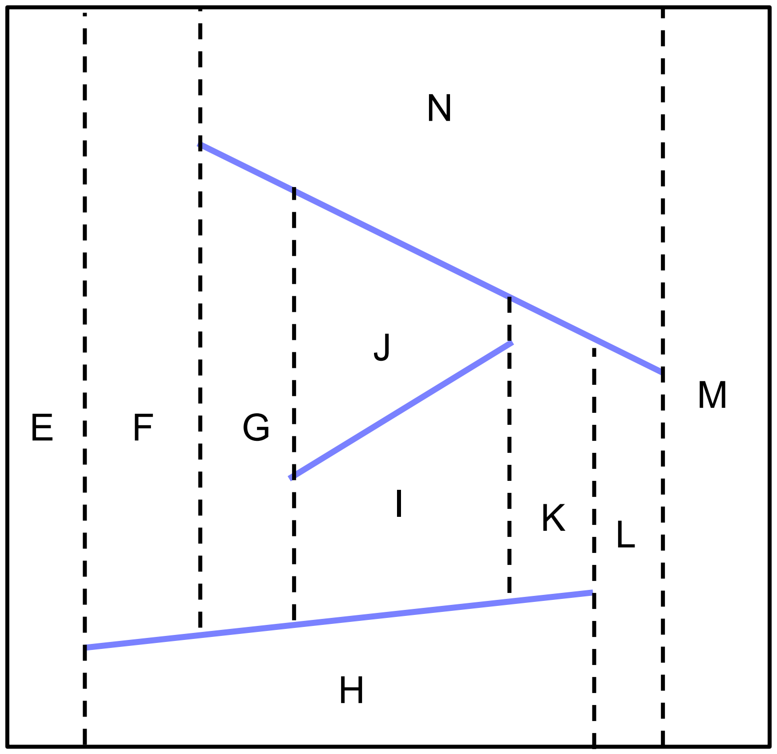
Q2 - Trapezoidal Map Computation



Q2 - Trapezoidal Map Computation

Ideas?

Sweep line

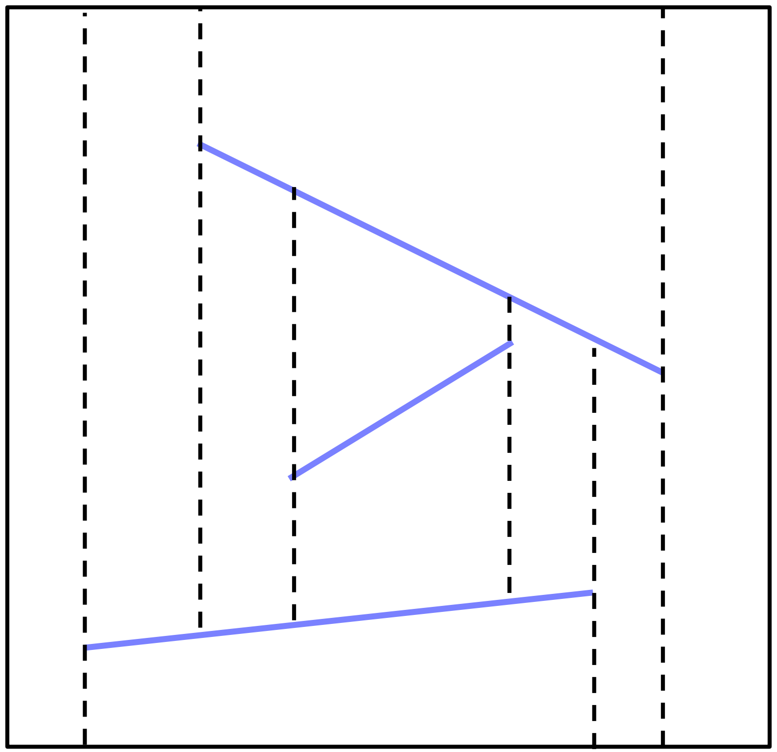


Q2 - Trapezoidal Map Computation

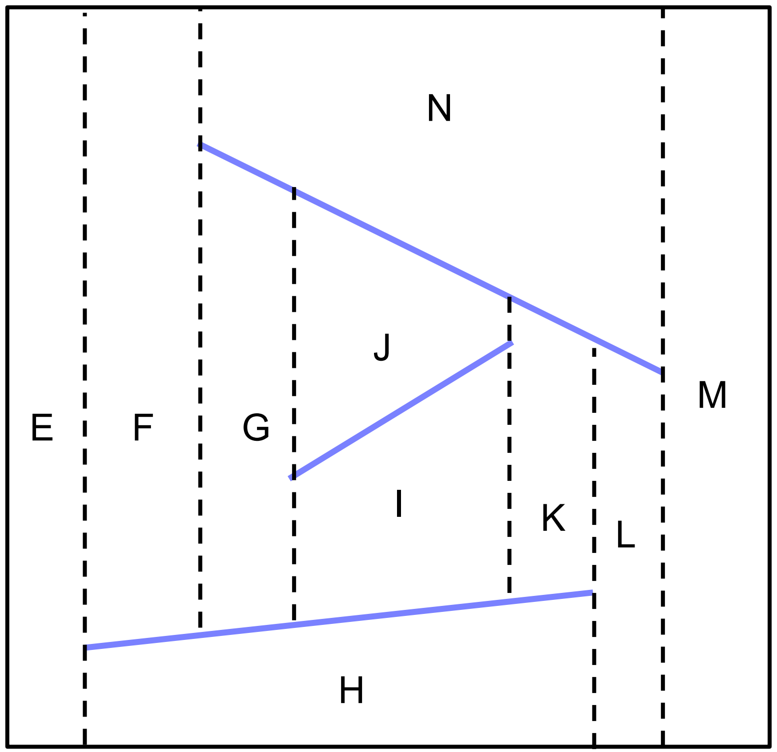
Solution: plane sweep algorithm.

- **Order:** From left to right.
- **Status:** The set of segments the sweep line cross.
- **Event handle:**
  - **Start and end:** Find the segment above and below, and add vertical lines to them, split the needed segments.
  - **Intersection:** swap the intersecting lines.

Q2 - Trapezoidal Map Computation



Q2 - Trapezoidal Map Computation



Plane sweep: $O((n+k)\ log\ n)$.

Q3 - Segment Intersection

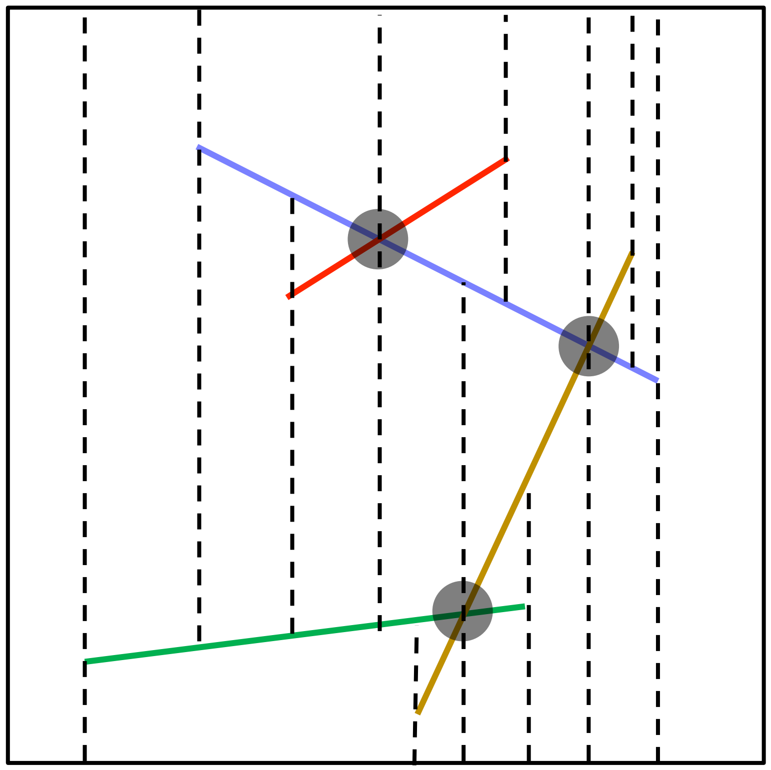
Given a set of segments, design an algorithm that computes all the intersections of pairs of segments.

- Complexity $O(n \log n + k)$ on average.

- Ideas?

Q3 - Segment Intersection

- In the trapezoidal map computation, we haven't handled intersecting segments.
- Each intersection point is part of the trapezoidal map → it will be handled at some point
  - we can report all the intersection as a byproduct.
- There exist an algorithm that compute the trapezoidal map for intersecting segments in $O(n \log n + k)$ on average



Q3 - Segment Intersection

Complexity:

- What is the average size of a trapezoidal map of n segments with k intersections?
- We can think as any intersection as 4 non-intersecting segments, thus the size is $O(n+k)$, and thus each point location and DAG update will take $O(\log(n+k)) = O(\log(n))$
- In addition to point locating, we will handle k intersections, thus, in total the complexity will be $O(n\log(n)+k)$ on average.