Line Segment Intersection

Given a set $S$ of $n$ segments:

- Do any two intersect?
- Which pairs intersect?
- How many pairs intersect?
- Find all intersection points
- Subdivide $S$ at intersections

Examples:

Given two polygons, compute intersection/union/convex hull

Given a polygon, is it simple?

Map overlay: Given two planar maps

Variants:
- Replace segments with lines, boxes, circles, curves, triangles in $\mathbb{R}^3$
Given two segments $pq$ and $rs$, do they intersect?

**Intersect lines $pq$ and $rs$**

$\text{orient}(p,q,r) = 0$
$\text{orient}(r,s,z) = 0$

$\begin{vmatrix} 1 & a & b \\ 1 & c & d \\ 1 & x & y \end{vmatrix} = 0$
$\begin{vmatrix} 1 & c & f \\ 1 & g & h \\ 1 & x & y \end{vmatrix} = 0$

$(b-d)x + (c-a)y = ad-bc$

Check if int point lies on segments

**Crossing $p$$r$ intersection is single point in interior of both segments**

$\text{orient}(p,q,r) \neq \text{orient}(p,q,s)$

$\text{orient}(p,r,s) \neq \text{orient}(q,r,s)$

$O(1)$ time

**Detecting intersections**

For new
Assume general position
- distinct coordinates
- no 3 collinear endpoints

Intuition:
Continuously sweep vertical line left to right
Maintain intersections between $S$ and $l$
Implementation:

Maintain sequence of segments crossing \( l \)

sorted by \( y \)-coord of intersection points

Changes at left and right endpoints

(and segment crossings)

Data structure? Balanced binary search tree

insert \( \mathcal{O}(\log n) \) time

delete \( \mathcal{O}(\log n) \) time

Use orient tests instead

of comparisons

\[ p_j \text{ above } p_k q_k \iff \text{orient}(p_k, q_k, p_j) > 0 \]

Init BST \( \leftarrow \emptyset \)

Consider endpoints in left-to-right order

sort: \( \mathcal{O}(\log n) \)

At each left endpoint \( p_i \):

\( i \leftarrow \text{pred}(p_i) \)

\( k \leftarrow \text{succ}(p_i) \)

insert \( j \) into BST

if \( p_i q_i \) crosses \( p_j q_j \) report True

if \( p_i q_i \) crosses \( p_k q_k \) report True

At each right endpoint \( q_j \):

delete \( j \) from BST

\( i \leftarrow \text{pred}(q_j) \)

\( k \leftarrow \text{succ}(q_j) \)

if \( p_i q_i \) crosses \( p_k q_k \) report True

return False

\( \mathcal{O}(\log n) \) time