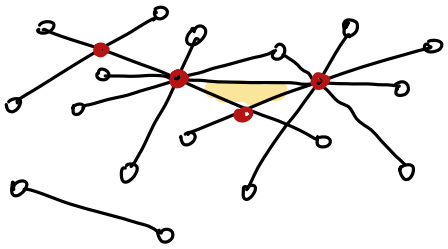
# Segment Intersection



Detection — Any intersections?

Counting — How many?

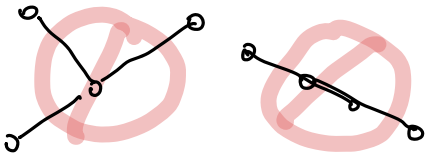
Reporting — List them

Construction — Build arrangement of input segments

Arrangement = planar straight-line graph

vertices = endpoints and intersection
edges = subsegments btwn verts

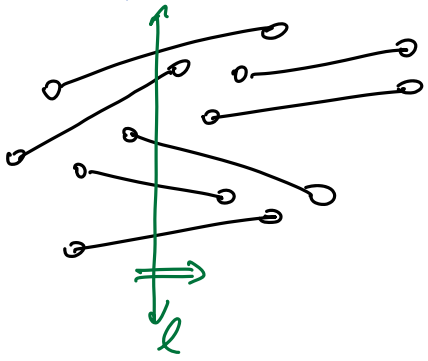
Necessarily considering some degeneracies



No 3 endpoints are collinear!

---

[Shamos-Hoey] Sweep to detect intersection




Sweep vertical line $\ell \Longrightarrow$

Maintain sequence of intersections between $S$ and $\ell$

Balanced BST — sweep structure

left end → insert
check 2 pairs
right end → deletion
check 1 pair

$O(n)$ events — $O(n \log n)$ time

[Bentley-Ottman] Count intersecting pairs

Also maintain a priority queue of events priority = x-coord
- left ends
- right ends
- intersections (pair of segments)

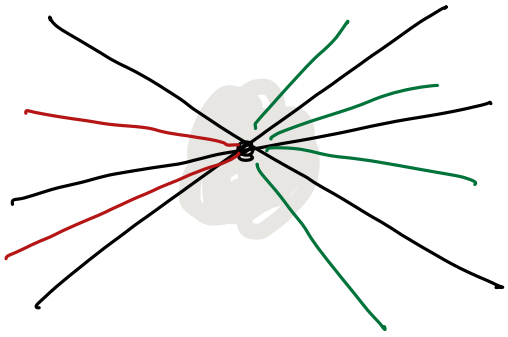
$O(n+k)$ events  $k$ = # intersecting pairs

$O(\log n)$ time for each

$\rightarrow O((n+k)\log n)$

Output sensitive IFF
- Reporting
- Gen. position.

---

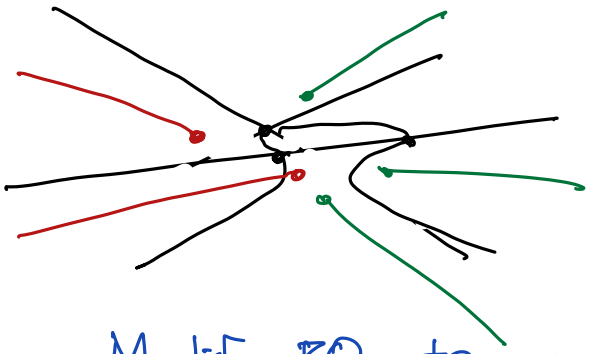


$2+4+\binom{3}{2} = 9$ events

BO algorithm implicity perterbs into general position

$\downarrow$

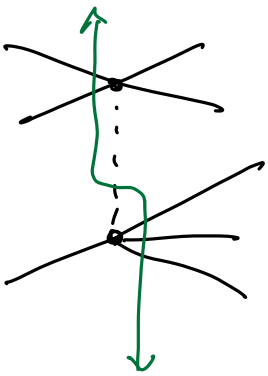
not necessarily consistent with a geometric perturbation (topological)



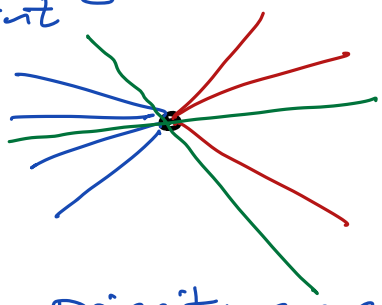
Inconsistencies can kill you later

Modify PQ to consider all events at same location to be a single event

priority = x-coord, ties broken by y-coord



Each event stores all segments thru that point
- Left
- Thru
- Right.



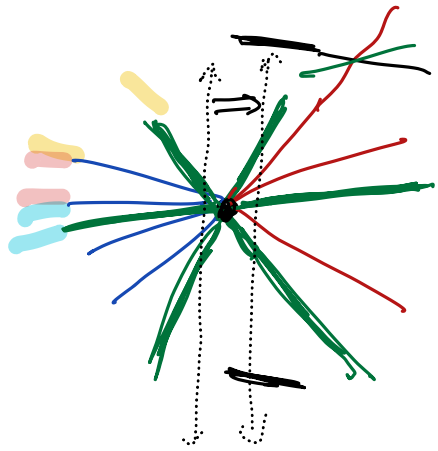
Use another balanced BST as a priority queue

Insert RtEnd(i) ← Insert (detect + handle duplicates)

Insert LEnd(i) ← Extract-Min

Insert Crossing(i,j)

ExtractMin $\longrightarrow$ event record $X$



Operations on sweep BST

Delete X.Left

Delete X.Thru

Insert X.Thru

in reverse order

$\longrightarrow$ increasing slope

Insert X.Right

Degree $d \Rightarrow O(d)$ operations on sweep dict. $\Rightarrow O(d \log n)$

+ check for intersections btwn

highest seg in X.Thru $\cup$ X.Right and succ

lowest seg ——————— pred

$\Rightarrow O(1)$ Inserts into event queue. $\Rightarrow O(\log n)$

+ $O(d)$ bookkeeping to build the arrgh graph

$$\text{Overall running time} = O(n \log n) + \underbrace{\left(\sum_v \deg(v)\right)}_{2E} O(\log n)$$

$$= O(E \log n)$$

$$= O(V \log n) = O((n + \overset{\#\text{int. pts.}}{I}) \log n)$$

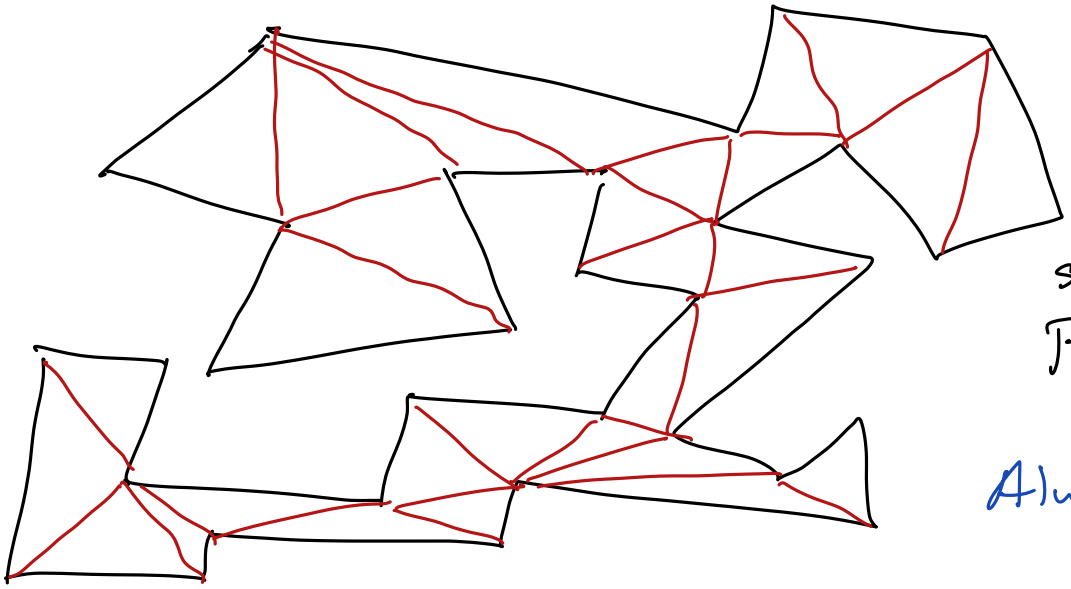
because arrgh graph is planar

Euler's theorem: $V - E + F = 2$

$\Rightarrow E \leq 3V - 6$

---

Other techniques $\Rightarrow O(I + n \log n)$ time



simple polygon triangulation

Always exists?