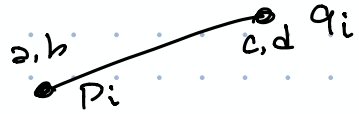
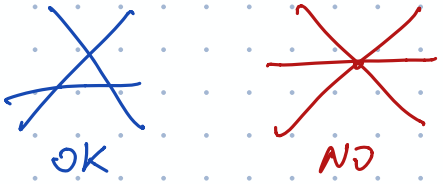
# Segment Intersection




Input: set $S$ of $n$ line segments $(a,b),(c,d)$

Output: All pairs $(i,j)$ s.t. $p_iq_i \cap p_jq_j \neq \emptyset$.

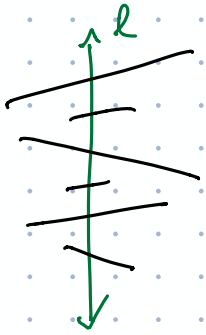



Assume general position

---

## Detecting Intersections — sweep line algorithm

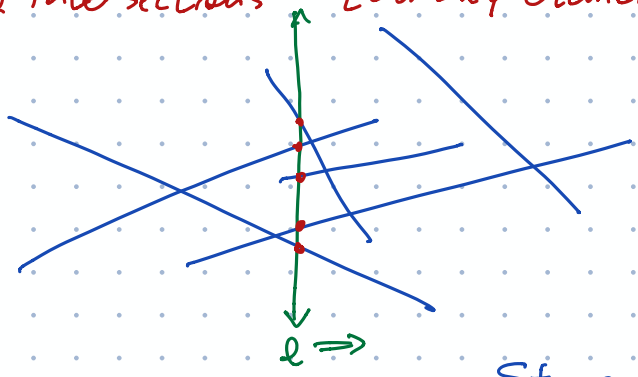
[Shamos Hoey 78]




Sweep structure:
BST of segments intersecting $\ell$
sorted vertically

Events: changes to sweep structure
Endpoints ← Sorted in advance

## Find All intersections [Bentley-Ottmann]




Same sweep structure

Events:
- Left endpts → Insert
- Right endpts → Delete
- Crossing pts → Swap!
  - ↑ detect on the fly

Store events in a priority queue
priority = x-coord

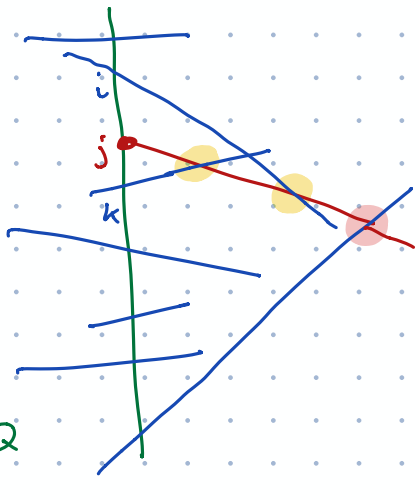
Insert
ExtractMin

---

Insert all endpts into PQ
while PQ is not empty
  get next event

① if left endpoint $p_j$
  - $i \leftarrow pred(j)$
  - $k \leftarrow succ(j)$
  - insert $j$ into BST

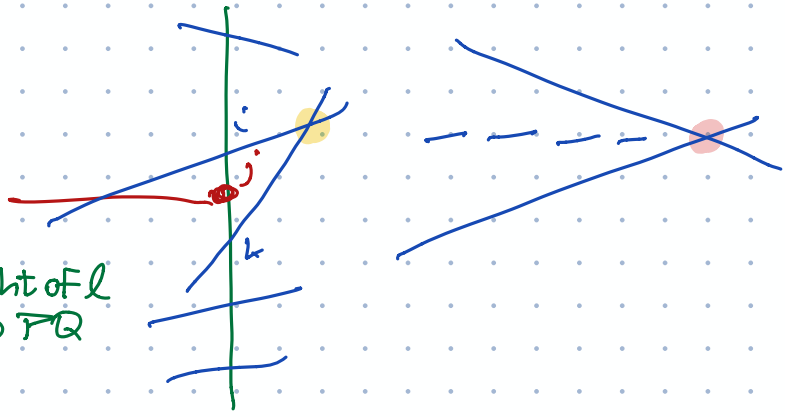
  if $s_i$ and $s_j$ cross right of $\ell$
    Insert crossing into PQ
  Similarly for $s_j$ and $s_k$



② if right endpoint $q_j$

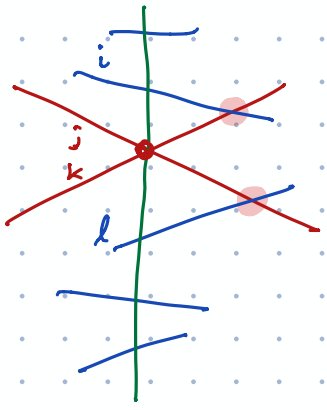
$i \leftarrow pred(j)$
$k \leftarrow succ(j)$
delete $j$ from BST

if $s_i$ and $s_k$ cross right of $\ell$
Insert crossing into PQ




③ if crossing $s_j \wedge s_k$
report crossing!
w log $j = pred(k)$
$i \leftarrow pred(j)$
$\ell \leftarrow succ(k)$

SWAP { delete $j$ ~~and $k$~~ from BST
insert $j$ ~~and $k$~~ into BST
break ties by slope



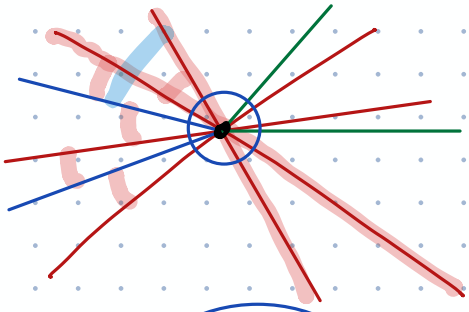

if $s_j$ and $s_\ell$ cross to right
insert crossing
if $s_k$ and $s_i$ cross to right
insert crossing

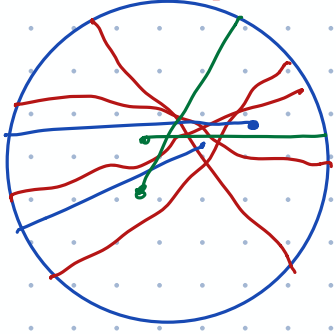
every event → O(1) BST ops, O(1) PQ ops } O(log n) time

Overall time = $O(n \log n + K \log n)$

where $K$ = # crossing pairs

---

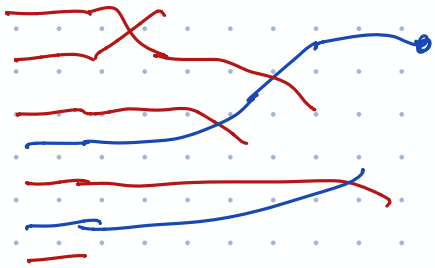




check for crossing on or right of $\ell$
x-coord of the current event

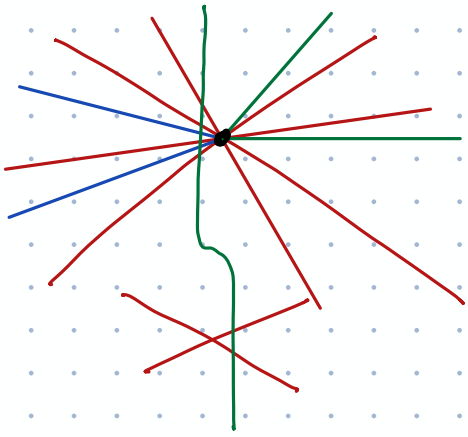
Coincident events are implicitly perturbed into g.p.

- $d$ segments ⇒ $\leq \binom{d}{2}$ crossing pairs
  $\leq \binom{d}{2}$ events
- Tie breaking may not be consistent with geometric perturbation
  ⇒ FUTURE CRASHES



## Building arrangement

V = endpoints and intersection pts
E = subsegments between vertices



**Modify PQ:**

priority = x-coord
break ties by y-coord

Each event record X stores all segments known to contain event point (x,y)
→ X.segments

Insert(i,j) — insert crossing between $s_i$ and $s_j$
$(x,y) = s_i \cap s_j$

Use balanced BST as PQ
→ O(log n) time per op

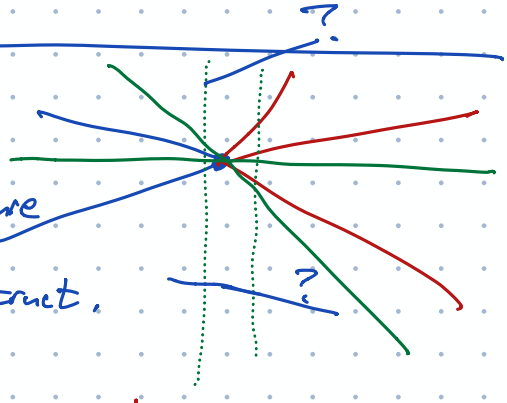
Find record X with priority (x,y)
insert new record if nec.

Add i and j to X.segments
↑ BST

X ← ExtractMin

Split X.segments → L, IZ, Thru

reversing order →
- Delete L and T from sweep structure
- Insert IZ and T into sweep struct.
  break ties by slope




Check for crossings btwn highest + pred
lowest + succ

O(d) BST operations
+ O(1) PQ operations
+ O(d) time to update output graph DS
→ O(d log n) time

where d = # X.segments

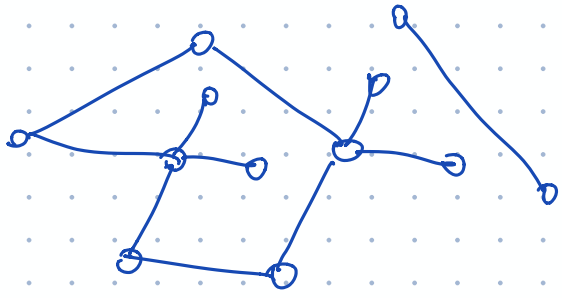
Overall time: $O(n \log n) + \sum_{i=1}^{V} O(\deg(v_i) \log n)$

$O(n \log n) + \boxed{\sum_{v} \deg(v)} \cdot O(\log n)$

$\sum_{v} \deg(v) = 2E$

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

$= O((n+V) \log n)$ by Euler's Formula!



$V - E + F \leq 2$

$\Downarrow$

$E \leq 3V - 6$

equality if every face is a triangle