Line segment intersection

Given : a set of segments $S := \{s_1, \cdots, s_n\}$

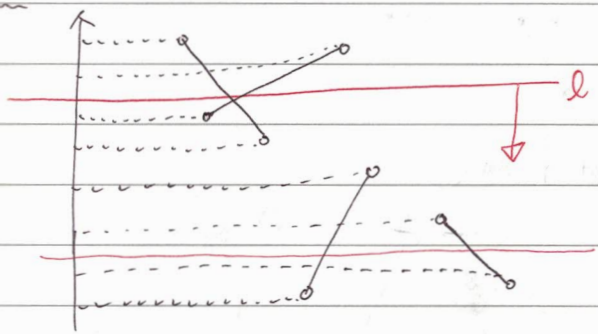
Goal : Identify all intersections of S.

(Assumption : ① No horizontal lines ② No three segs intersects at a point (can be lifted)
③ No two segs intersects w/ positive volume. → ? distinct curve in practice)

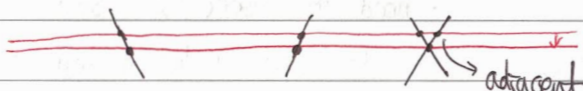
Naive alg : $O(n^2)$ time. (Worst case, $\Omega(n^2)$ intersections.

What if # intersection is subquadratic? Want $\tilde{O}(n+I)$ time, (I = # intersections)

Idea 1. y-coord intervals of S



Idea 2. Ordered list of the segs intersecting w/ ℓ.



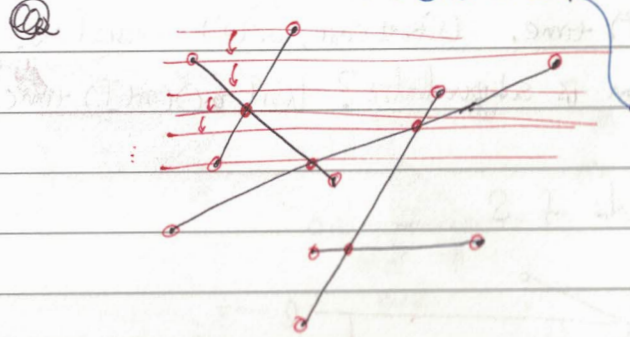
Issue 1. Consider only "meaningful" timesteps.

Issue 2. Maintain the ordered list of the segs intersecting w/ ℓ.

~~Balanced binary search trees~~

For issue 1, event queue Q implemented by BBSTs
issue 2, status $\mathcal{T}$ (w/ some minor adjustments)

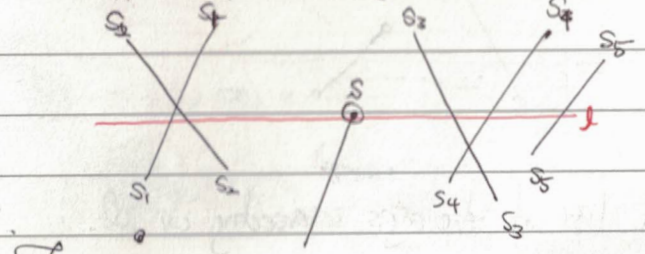
Event points = upper & lower endpts of all segs
$\cup$ all intersections



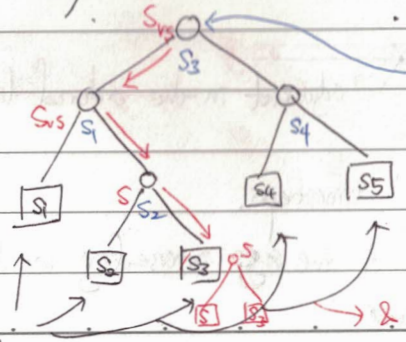
engqueued from beg.

on the fly!

Event 1. Upper endpt of a seg.



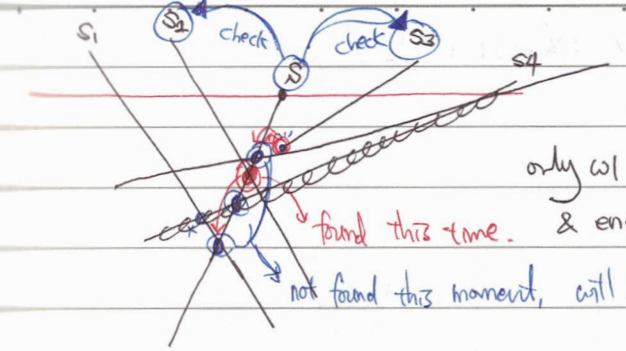
$\mathcal{T}$.



- need to insert "s" btw s_2 & s_3!
  the rightmost leaf of left subtree
  → we can then calculate the right position of s.
  in $O(\lg n)$ time.

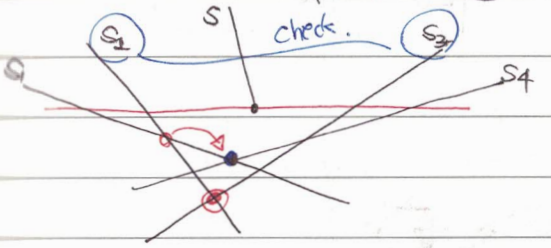
→ & rebalancing.

the segs intersecting w/ l



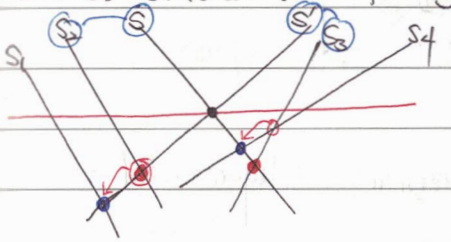
check whether s intersects only w/ its adjacent segs (s_2 & s_3).
& enqueue the intersections into Q

Event 2. Lower endpt of a seg.



enqueue the intersection into Q &
update $\mathcal{T}$. (delete s)

Event 3. Intersection of segs.



enqueue the intersections into Q &
update $\mathcal{T}$ (swap s & s')

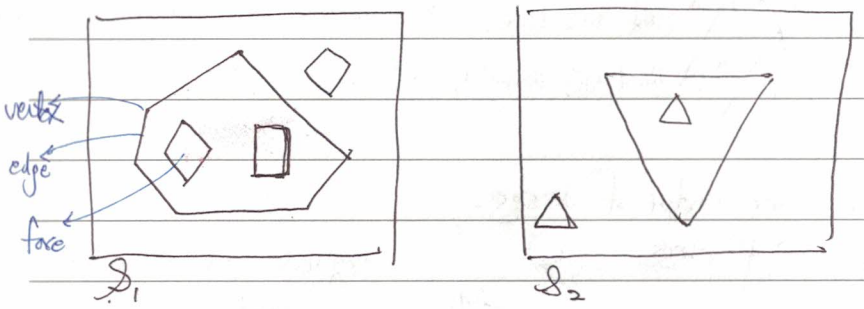
- Plane sweep alg

Running time?

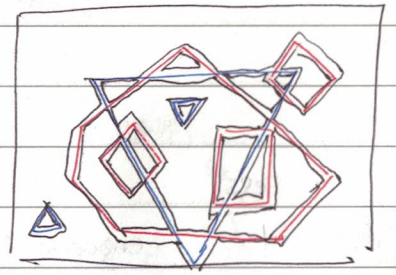
- At the beg, init Q takes $O(n \lg n)$ time
  $\mathcal{T}$ constant time
  $I = O(n^2)$
- At each event, update $\mathcal{T}$ in $O(\lg n)$ time $O(\lg(n+I))$
  enqueue at most 2 events into Q. $O(\lg n)$ time

#events = $c(n+I)$ $\Rightarrow$ $O((n+I)\lg n)$.

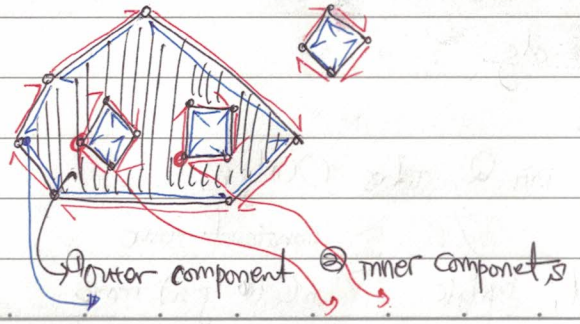
Subdivision overlay



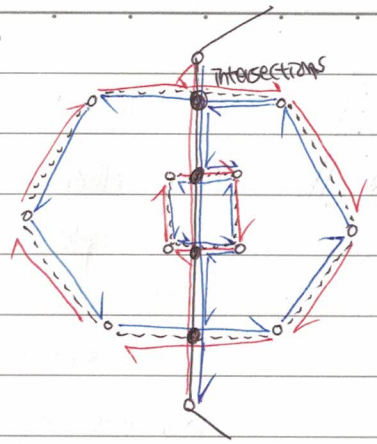
↓ overlay



How to represent/compute a subdivision? Doubly-connected edge list.



Overlay?

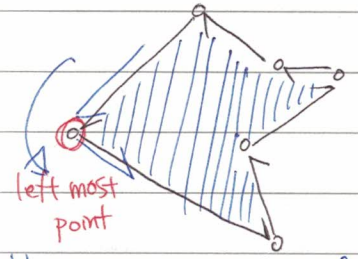


Alg

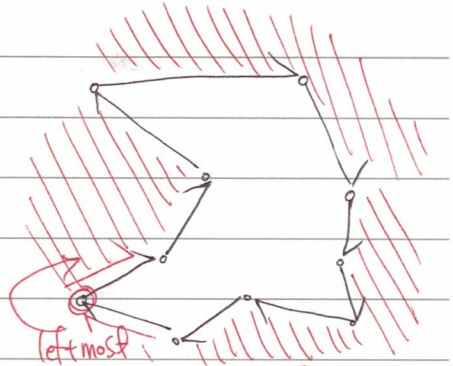
From the doubly-conn edge lists of S_1 & S_2, find all intersections and reconnect the edges.

Face?

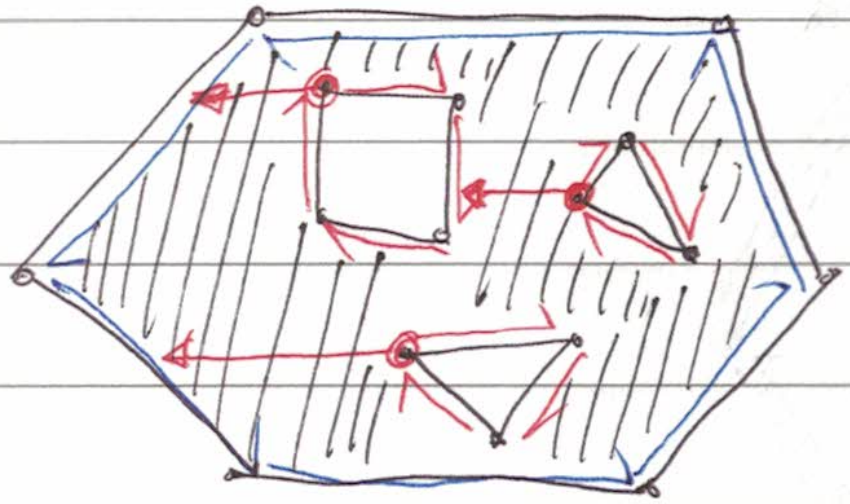
- Given a doubly-conn edge list, determine the outer comp & inner comps of each face?



the outer comp of a face



an inner comp of a face



for each inner cycle
find another inner cycle
or outer cycle that is
immediately on the left.
from its leftmost vertex.

→ These form inner comps of
the face of the outer comp.

During @ plane sweep alg, as we maintain T,
it is easy to find its left-immediate cycle.