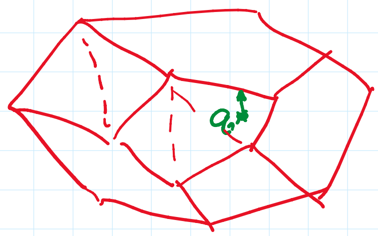
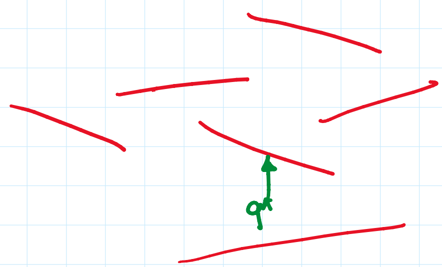
Problem: Point Location



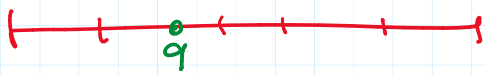
by Euler's formula,
edges/faces = O(n)

given planar subdivision with n vertices
build data structure s.t.
given query pt q, find region containing q
equiv: find line segment immediately above q

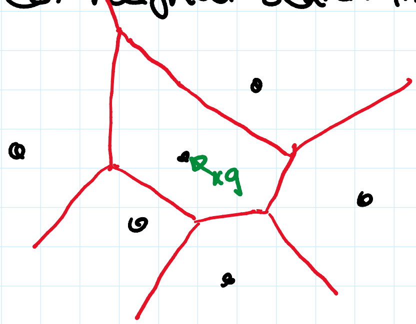
⇒



1D:

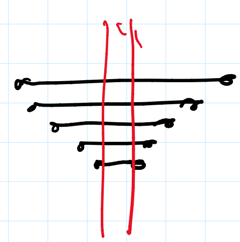


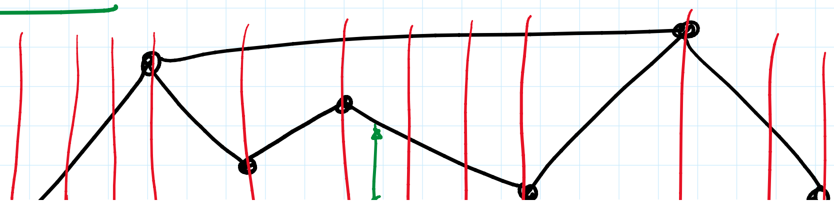
Appl: nearest neighbor search in 2D

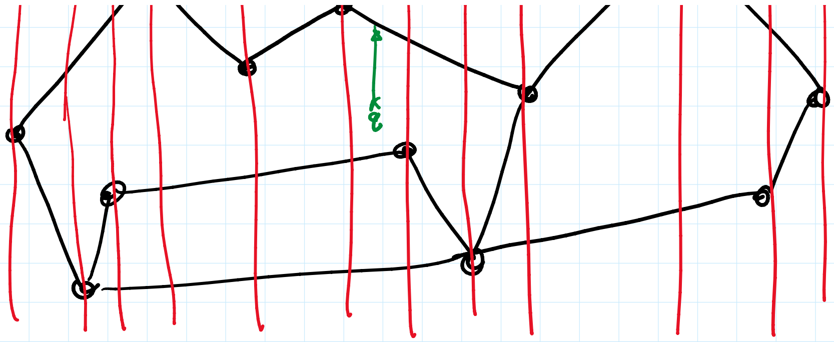


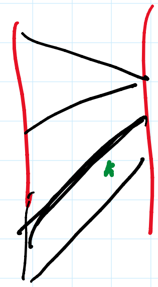
Voronoi diagram

Method 0:









divide into n vertical slabs
store y-sorted list in each slab

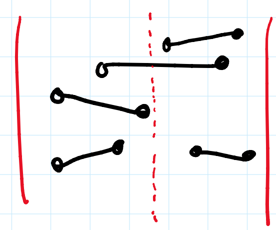
$\Rightarrow$ query time $O(\underbrace{\log n}_{\text{binary search in } x} + \underbrace{\log n}_{\text{binary search in } y}) = \boxed{O(\log n)}$

Space $\boxed{O(n^2)}$
preproc time $\boxed{O(n^2 \cancel{\log n})}$

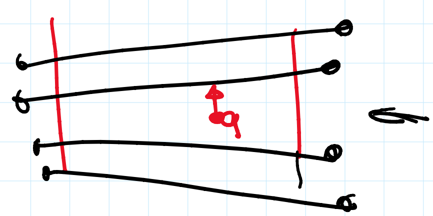
Method 5: Segment Tree

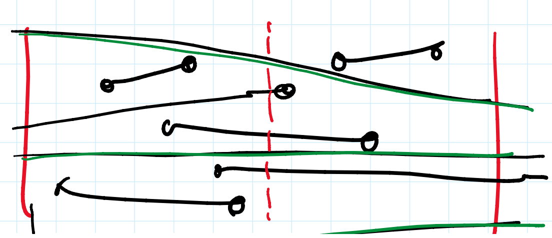
given n disjoint line segments intersecting slab σ,

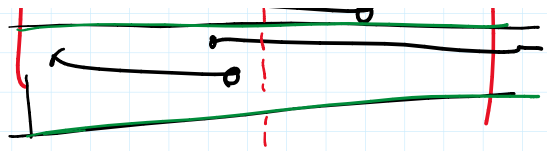
divide by median x
remove all long segs in σ
& store them in y-sorted list
recurse in left & right

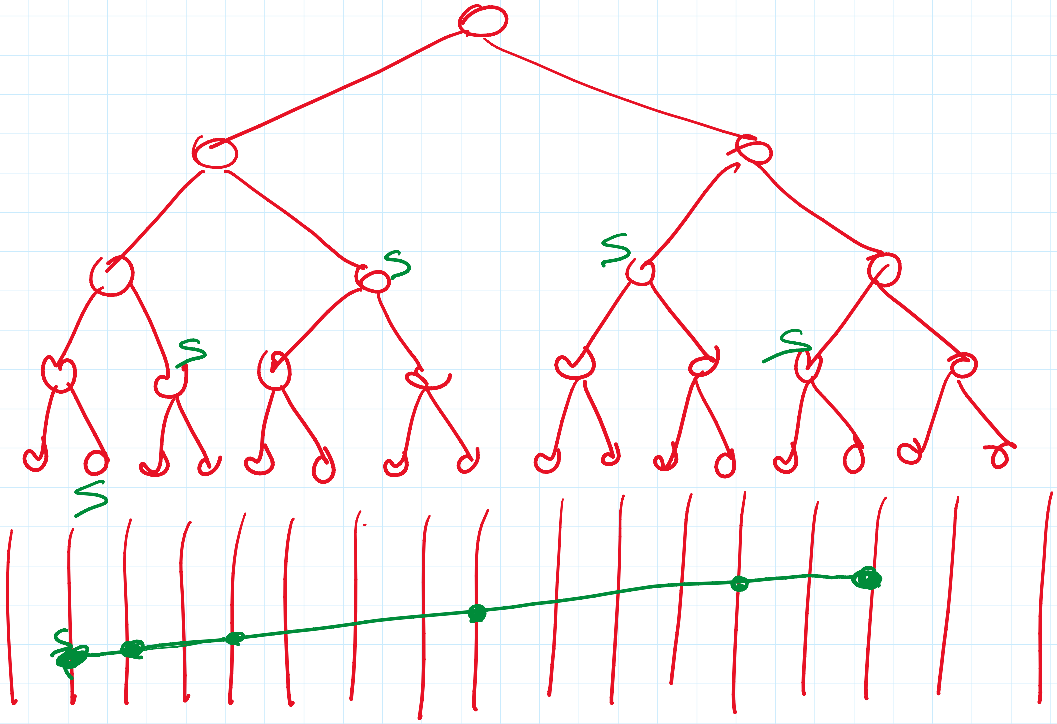


Def segment S is long in σ
if it completely cuts across σ.









each seg s is stored $O(\log n)$ times

$\Rightarrow$ space $\boxed{O(n \log n)}$

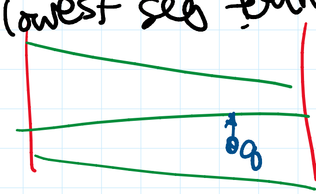
preproc time $\boxed{O(n \log^2 n)}$

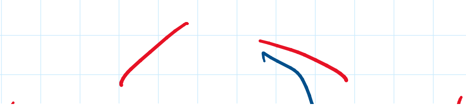
Query algm, for query pt q:

find long seg immediately above q by binary search

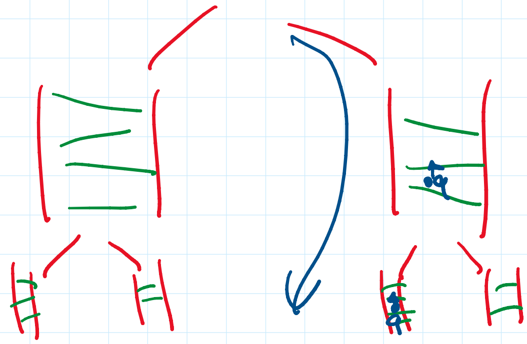
if q left of median x
recurse left
else recurse right

return lowest seg found.





binary searches
at $O(\log n)$



binary searches at $O(\log n)$ nodes along a path

$\Rightarrow \boxed{O(\log^2 n)}$ query time

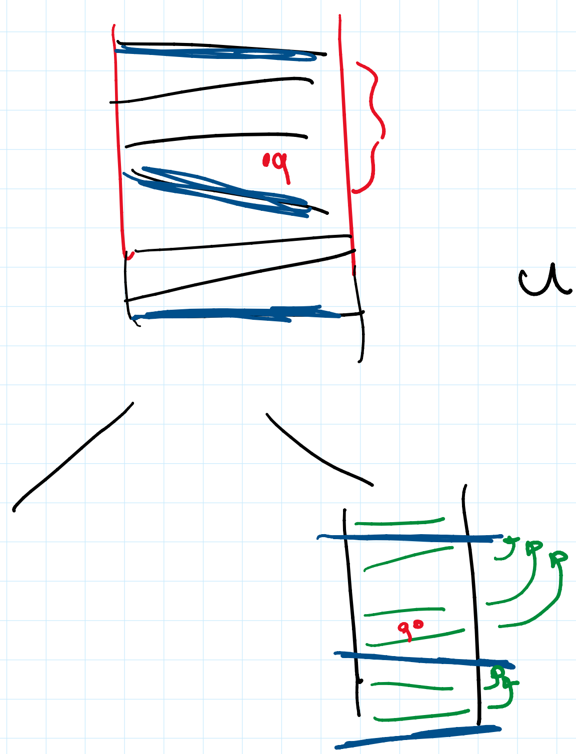
How to speed up query?

issue: parent list & child list not related...

Method 2: Segment Tree + "Fractional Cascading"

(Chazelle, Guibas '86)

idea: pass a $\frac{1}{b}$ fraction of parent list to child list



$b=3$

u

v

Let $L(u)$ = original y-sorted list at node u

Will generate an "augmented list" $L^+(u)$

Define sample(L) = sublist of L formed by taking 1 out of every b elem

for each child v of u,

let $L^+(v) = L(v) \cup \text{sample}(L^+(u))$

store succ ptrs between $L^+(v)$ and sample($L^+(u)$)

⋮