Method 3: Trapezoid Tree (Preparata '81)

use a tree where cells are vertical trapezoids instead of slabs

Given n segs intersecting trapezoid τ,

if no long segs,

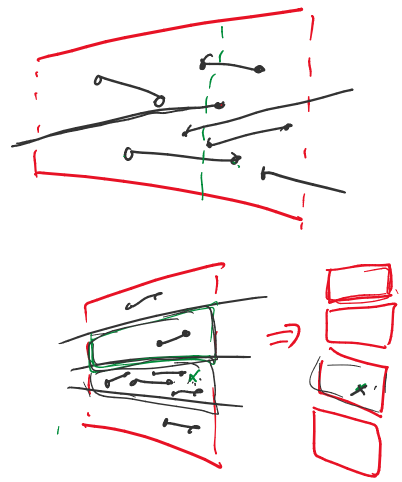
divide τ by median x

degree-2 node

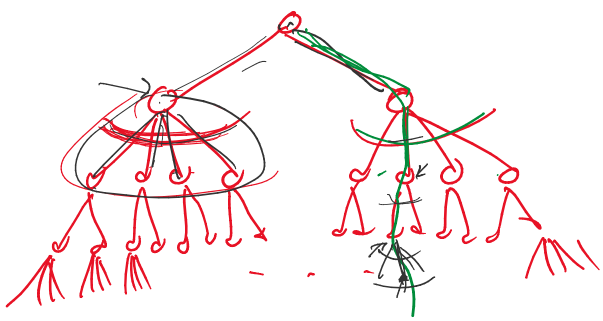
else

divide τ by all long segs

multi-degree node



do weighted search here



$O(\log n)$ height

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

query time $O(\underbrace{\log n}_{\#\text{ levels}} \cdot \underbrace{\log n}_{\text{binary search at each multi-deg node}}) = \boxed{O(\log^2 n)}$

Lemma Given m elements $y_1, \ldots, y_m$ in 1D, with weights $w_1, \ldots, w_m \geq 1$, $W = \sum_{i=1}^{m} w_i$

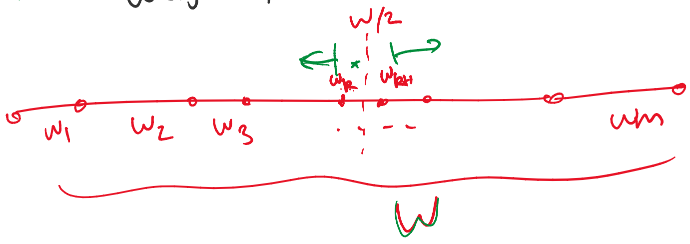
can find pred y_i of any query pt q in time $O(\log \frac{W}{w_i} + 1) = O(\log W - \log w_i + 1)$

can find pre...

in time $O\left(\log \frac{W}{w_i} + 1\right) = O(\log W - \log w_i + 1)$

comps $1 \cdot (\log W - \log w_i) + O(1)$.

Pf sketch: weighted/biased binary search



□

at each multideg node,
use lemma, with weight(τ) = # leaves in τ's subtree

$$\Rightarrow \text{total query time } O\Big(\underbrace{\log W_0 - \cancel{\log W_1} + 1 + \cancel{\log W_1} - \cancel{\log W_2} + 1 + \cancel{\log W_2} - \log W_3 + 1 + \cdots}_{\log n \text{ terms}}\Big)$$

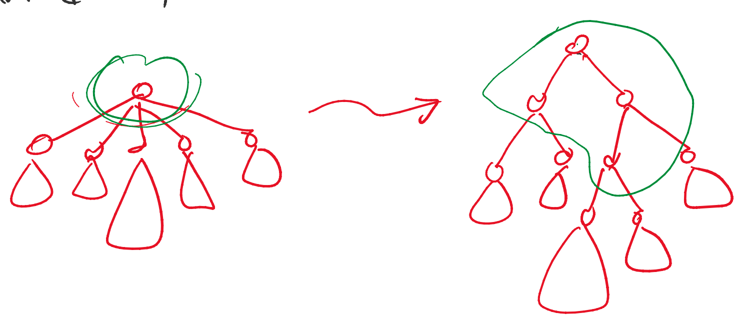
telescoping sum

$$= O(\log W_0 + \log n)$$

$W_0 = O(n \log n)$

$$= \boxed{O(\log n)}$$

Rmk - can directly convert tree to binary

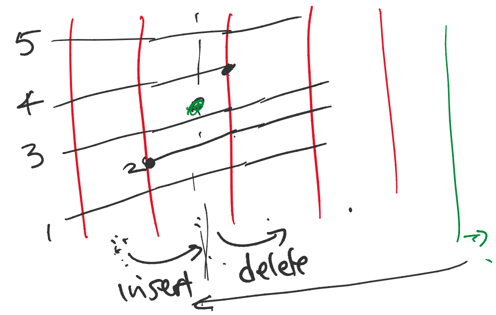


- related to BSP tree (binary space partition)

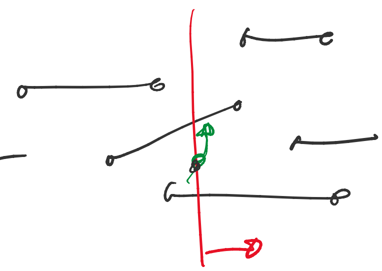
Rmk - Seidel-Adamy '98: any tree data structure (in some model) for 2D point location requires $\Omega(n \log n)$ space

Method 4: Persistent Search Tree (Sarnak, Tarjan '86)

back to Method 0 (slab method)



Sweep from left to right
maintain y-sorted list L
if we hit left endpt,
  insert to L
if right endpt,
  delete from L

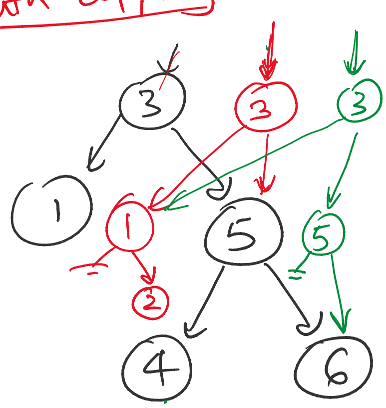


can use balanced search tree for L
insert/delete/search in $O(\log n)$ time

to answer query,
need to do pred search in a past version of L

persistence - remember history s.t. we can query in past

One implementation of persistent search tree:
path copying



insert(2)
delete(4)

rotations are similar }
(can avoid by pre-sorting all segs by y)

④ ⑥ (can avoid by pre-sorting all segs by y)

query time $O(\log n + \log n) = \boxed{O(\log n)}$

init binary search in time i.e. x-coord

query in one search tree

Space $\boxed{O(n \log n)}$

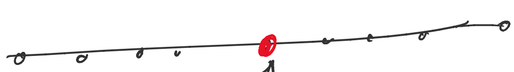
preproc time $\boxed{O(n \log n)}$

Rmk - Sarnak-Tarjan improves space to $O(n)$ by limited path copying

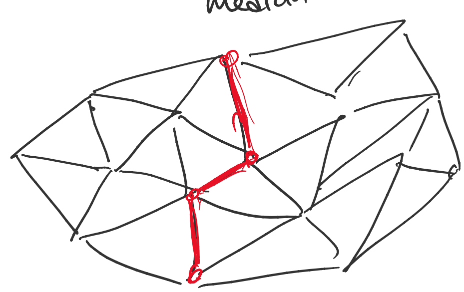
Rmk - technique general (bounded degree property)

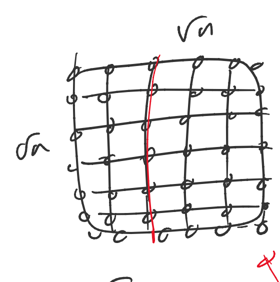
Method 5: Planar Separators (Lipton-Tarjan '77)

1D



2D





Thm Given a triangulated planar graph G with n faces, can find subset R of $\boxed{O(\sqrt{n})}$ edges that divide G into 2 regions each with $\leq \frac{2}{3}n$ faces.