Method 3: Trapezoid Tree (Preparata '81)

Use a tree where cells are trapezoids instead of slabs.

Given \( n \) segs intersecting trapezoid \( T \),
if no long segs,
divide \( T \) by median \( x \)
else
divide \( T \) by all long segs

\[ \Rightarrow O(n \log n) \text{ space as before} \]
query time \( O(\log n \cdot \log n) = O(\log n) \)

lemma
Given \( m \) elements \( y_1, \ldots, y_m \) in 1D,
with weights \( w_1, \ldots, w_m \geq 1 \), \( W = \sum w_i \)

Can find \( \text{pred} y_i \) of any query \( p \) in time \( O(\log \frac{W}{m} + 1) = O(\log W - \log m + 1) \)
Can find your \( z \) in time \( O \left( \log \frac{W}{w} + 1 \right) = O \left( \log W - \log w + 1 \right) \)

**Proof Sketch:** weighted/biased binary search

At each multidegree node, use lemma, with weight \( \text{weight}(T) = \# \) leaves in \( T \)'s subtree

\[
W_0 = O(n \log n)
\]

\[
= O \left( \log W_0 + \log n \right)
\]

\[
= O \left( \log n \right)
\]

**Remark:** can directly convert tree to binary

- Related to **BSP** tree (binary space partition)
Runk - Seidel-Adamy’98: any tree data structure (in some model) for 2D point location requires $O(n \log n)$ space

Method 4: **Persistent Search Tree** (Samak, 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 prev search in a past version of $L$

**Persistence** - remember history st.
  - we can query in past

One implementation of persistent search tree?
  - path copying

Rotations are similar
(can avoid by pre-sorting all segs by y)
query time \( O(n \log n + \log n) = O(n \log n) \)

Space \( O(n \log n) \)

Preproc time \( O(n \log n) \)

(Rank): Sanak-Tarjan improves space to \( O(n) \) by limited path copying

(Rank): Technique general

(bounded degree property)

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

1D

2D

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