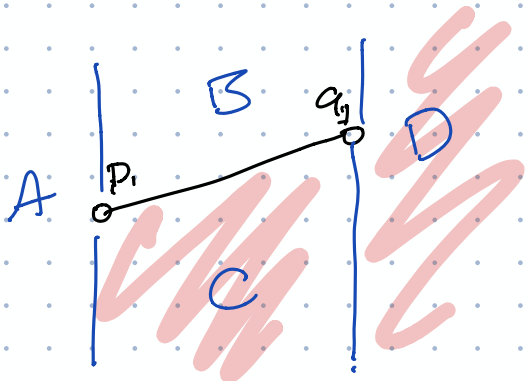
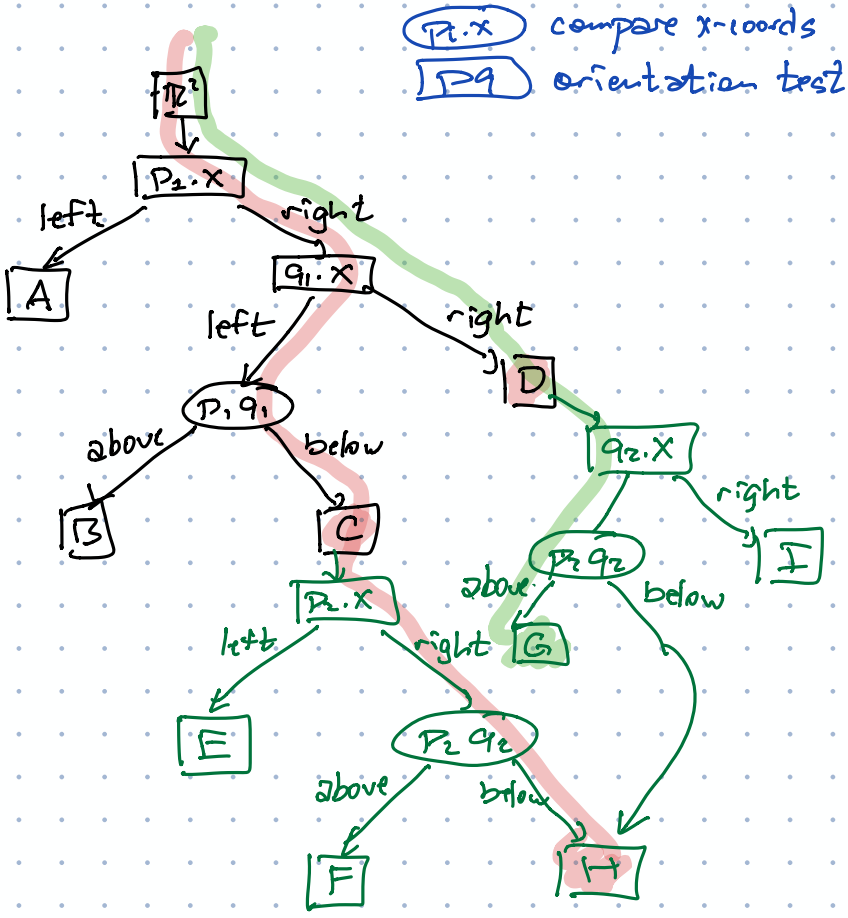
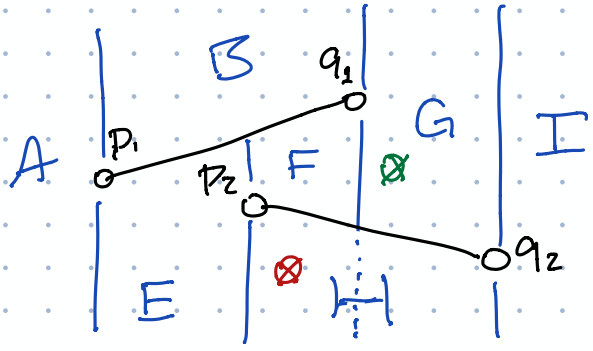
Joan Armatrading
"Down to zero"
Alabama Shakes
"Don't Wanna Fight"

Method #2 : History dag

Build a data structure to locate
an arbitrary point p in our decomposition







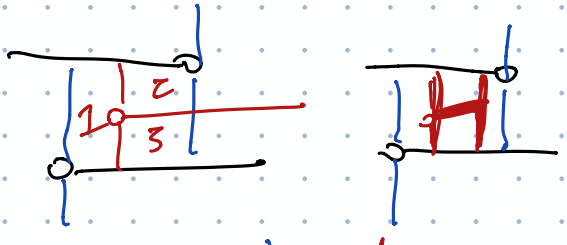
Point-location query:

Walk from root down to leaves
guided by x-comparisons and orient-tests

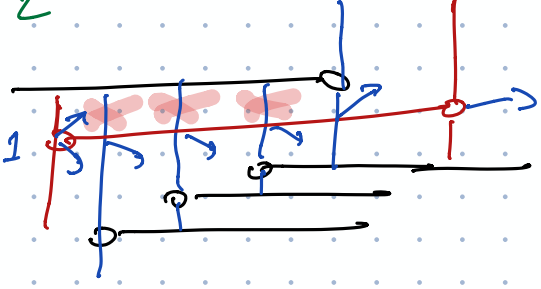
Time = O(path length)

Inserting a segment

Split trap t → 2, 3, 4 traps
↳ Add 1, 2, or 3 nodes below
the node representing t



Merge traps
↳ Merge corresponding nodes
in history dag



Space: $E[\#\text{nodes added by } i^{th} \text{ segment}]$

= Update time $\leq 3 \cdot E[\#\text{trapezoids in } T_{i-1} \text{ that intersect } p_i q_i]$

$\leq O(E[\#\text{trapezoids in } T_i \text{ that touch } p_i q_i])$

$= O(1)$

$E[\text{Space}] = O(n)$

Time: Let $t_i(p)$ = trapezoid in T_i containing p

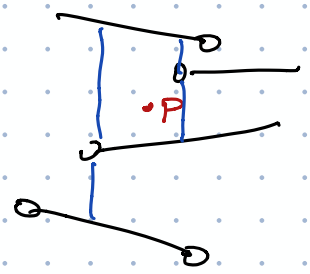
$E[\#\text{steps to locate } p]$

$\leq 3 \cdot E[\#\text{ times trapezoid containing } p \text{ changes}]$

$\leq 3 \cdot \sum_{i=1}^{n} Pr[t_i(p) \neq t_{i-1}(p)]$

$\leq 3 \cdot \sum_{i=1}^{n} Pr[p_i q_i \text{ touches } t_i(p)]$

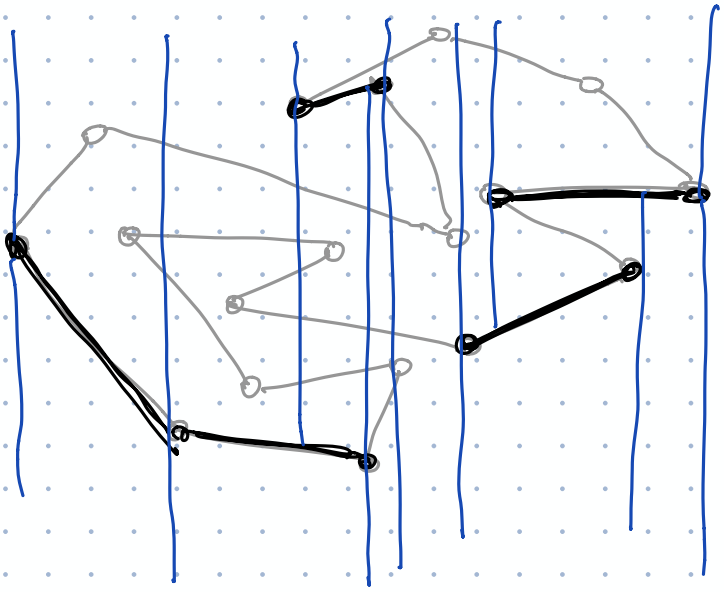
$\leq 3 \cdot \sum_{i=1}^{n} \frac{4}{i} = 12 H_n = O(\log n)$



Counting intersecting pairs $\longrightarrow$ $O(n \log n + k)$ exp time

Sweep: $O(n \log n + k \log n)$

Faster polygon triangulation



Spose we stop RIC after r segments

Point location via history $\Rightarrow O(\log r)$ exp. time

But we can locate all endpoints by walking around the polygon

$E[\text{Time}] = O(n + E[\#\text{crossings}])$

↑ polygon edges and vert walls

Claim: $E[\#\text{crossings}] \leq 4(n-r)$

Proof: Let R be random subset of r segments/edges

For any $s \in R$

$$\text{end}(s, R) = \#\text{ vert. walls in } T(R) \text{ that end on } s$$

For any $s \notin R$

$$\text{cross}(s, R) = \#\text{vert walls in } T(R) \text{ that cross } s$$

(1) $\sum_{s \in R} \text{end}(s, R) \leq 4r$

(2) $\text{cross}(s, R) = \text{end}(s, R \cup \{s\})$

$$E[\#\text{crossings}] = \mathop{E}_{|R|=r}\left[\sum_{s \notin R} \text{cross}(s, R)\right]$$

$$= \frac{1}{\binom{n}{r}} \sum_{|R|=r} \sum_{s \notin R} \text{cross}(s, R)$$

$$= \frac{1}{\binom{n}{r}} \sum_{|R|=r} \sum_{s \notin R} \text{end}(s, R \cup \{s\}) \quad \text{(2)}$$

$$= \frac{1}{\binom{n}{r}} \sum_{|R'|=r+1} \boxed{\sum_{s \in R'} \text{end}(s, R')} \quad \text{(1)}$$

$$\leq \frac{1}{\binom{n}{r}} \sum_{|R'|=r+1} 4(r+1)$$

$$= \frac{1}{\binom{n}{r}} \binom{n}{r+1} 4(r+1)$$

$$= \frac{r!(n-r)!}{n!} \cdot \frac{n!}{(r+1)!(n-r-1)!} 4(r+1) = 4(n-r)$$

IF we already know $t_r(p)$

Finding $t_i(p)$ via history

takes $O(1) \cdot \sum_{j=r+1}^{i} \frac{1}{j} = O(H_i - H_r) = O(\log \frac{i}{r})$

Faster algorithm [Seidel]

Choose integers $0 = r_0 < r_1 < \cdots < r_l = n$

```
Init
Randomly permute edges of P
For h ← 1 to l
    for i ← r_{h-1}+1 to r_h
        locate p_i via history
        trace p_i q_i and update
    if h < l
        locate all verts by walking along P
```

Time per segment: $\log\left(\frac{r_h}{r_{h-1}}\right)$

locate p_i via history / trace $p_i q_i$ and update → $O(1)$

walking along P → $O(n)$

	Pt loc	
$r_0 = 0$		
$r_1 = \frac{n}{\log n}$	$O(\log n)$	$O(n)$
$r_2 = \frac{n}{\lg \lg n}$	$O(\log \log n)$	$O(n)$
⋮	⋮	⋮
$r_h = \frac{n}{\lg^{(h)} n}$	$O(\log^{(h)} n)$	$O(n)$

phases $= O(\log^* n)$

Total expected time $= O(n \log^* n)$