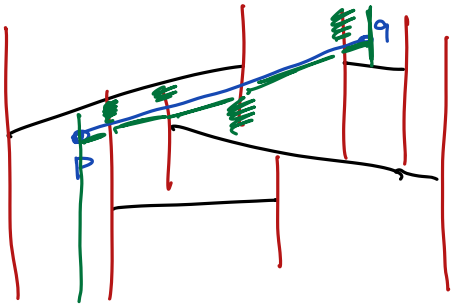
Randomized incremental construction

Start with empty structure

Add segments in random order
update at every addition



① Locate p
- Conflict lists / also updated with each new segment
- History dag

$O(\log n)$

$O(1)$
② Trace pq, splitting traps
③ Merge

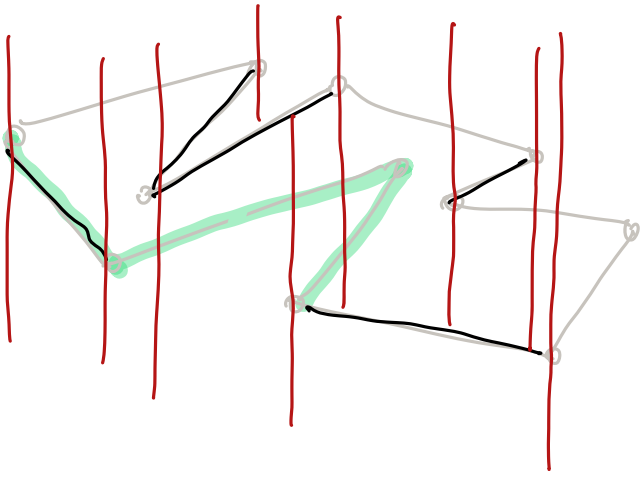
Backward analysis: imagine running the algorithm

$\Rightarrow O(n \log n)$ expected time

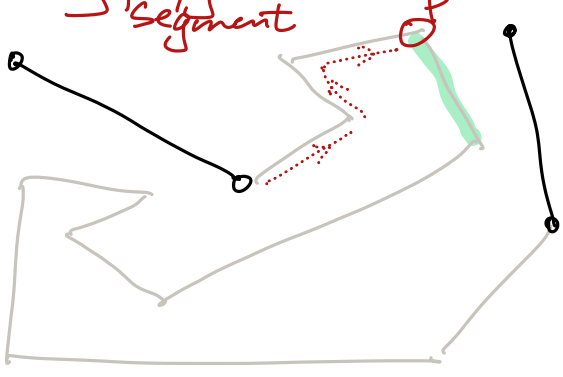
$O(n \log n)$ time whp

Polygon triangulation

Trap. decomp $+ O(n)$ time



We can locate p by walking along polygon from another segment



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

$\log^* n = \min\{k \mid \underbrace{\log \log \log \cdots \log}_{k \text{ logs}} n < 2\}$

$2^{2^{2^{\cdots^2}}}$ (n twos) $\uparrow\uparrow n$

[Seidel] [Clarkson Shor] [Devillers]

Alternate RIC + tracing entire polygon

Start by inserting r random edges of $P \to T_r$

Then locate every vertex by tracing P

$O(n)$ + [# crossings btwn P and walls of T_r]

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

Proof:

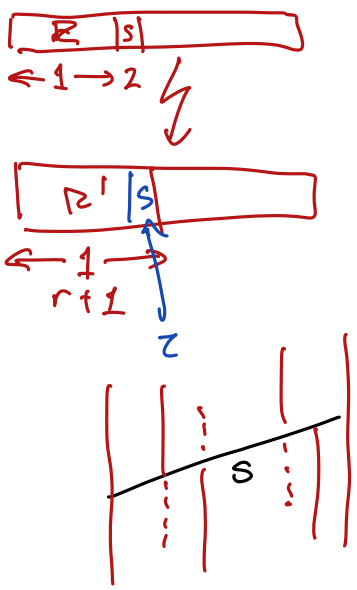
Notation: R = random subset of r edges

$\#\text{cross}(s,R)$ = # times segment $s \notin R$ crosses walls of $T(R)$

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

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

$E_s[\#\text{cross}] = \frac{\sum_s \#\text{cross}}{\#s}$



$$= \underset{|R|=r}{E}\left[\underset{s \notin R}{E}\left[\#\text{cross}(s,R)\right]\right] \cdot (n-r)$$

$$= \underset{|R'|=r+1}{E}\left[\underset{s \in R'}{E}\left[\#\text{cross}(s,R'-s)\right]\right] \cdot (n-r)$$

$$= \underset{|R'|=r+1}{E}\left[\underset{s \in R'}{E}\left[\#\text{end}(s,R')\right]\right] \cdot (n-r)$$

$$\le \underset{|R'|=n}{E}\left[\frac{2 \cdot 2(r+1)}{r+1}\right] \cdot (n-r)$$

$$= 4(n-r)$$

FastTrap(P): Define $0=r_0<r_1<r_2<\cdots<r_\ell=n$

randomly permute edges of $P \to s_1 \ldots s_n$

for $h \leftarrow 1$ to ℓ

  for $i \leftarrow r_{h-1}+1$ to r_h

    locate p_i via history $\leftarrow O\left(\log\left(\frac{n}{r_{h-1}}\right)\right)$

    trace s_i

  locate all verts by tracing P $\leftarrow O(n)$

If we already know $T_r(P)$

computing $T_n(P)$ takes

$$O(1)\cdot\sum_{i=r+1}^{n}\frac{1}{i} \text{ time}$$

$$= O(H_n - H_r) = O(\log n - \log r)$$

$$= O\left(\log\left(\frac{n}{r}\right)\right)$$

$r_1 = n \to$ standard RIC

$O(n\log n)$ pt loc

$O(n)$ tracing

$r_i = 2^i \to$ $O(n)$ pt loc

$O(n\log n)$ tracing

$r_1 = \frac{n}{\log n} \to$ First phase: $O(r_1 \log n)$ pt loc

$O(n)$ tracing

$r_2 = \frac{n}{\log\log n}$ ← Phase 2: $O\left(\log\left(\frac{n}{r_1}\right)\right) = O(\log\log n)$ per pt loc.

$r_3 = \frac{n}{\log\log\log n}$

$r_i = \frac{n}{\log^{(i)} n}$

$O(\log^* n)$ phases

$O(n)$ expected time each

$\Rightarrow O(n \log^* n)$ expected time