October 20, 2023 10:48 AM

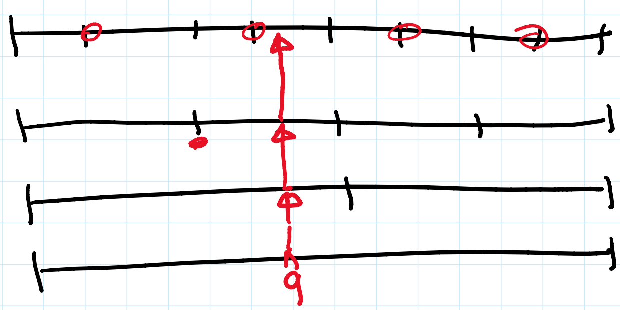
Point Location

segment tree + fractional cascading — $n \log n$ space, $\log n$ query

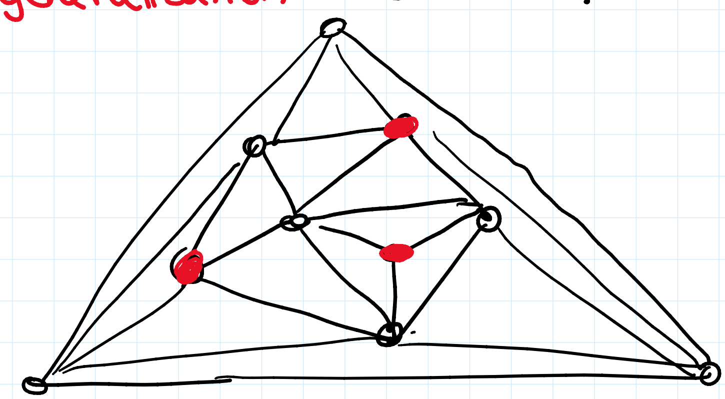
persistence — $O(n)$ space, $O(\log n)$ query

Method 4: Kirkpatrick's Hierarchy ('81)

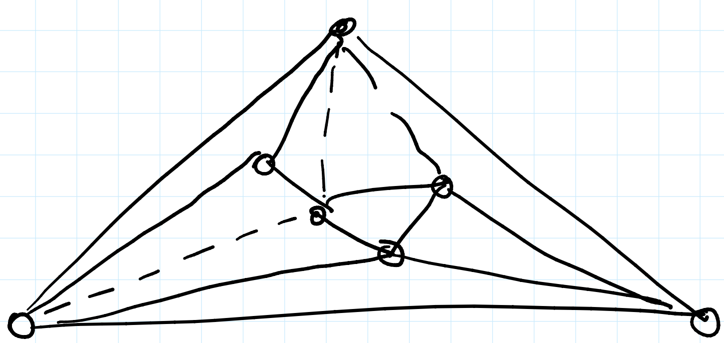
1D idea - remove half



2D generalization - remove a fraction



$\Downarrow$



$\Downarrow$

Preproc(S): // given triangulation S

repeat {

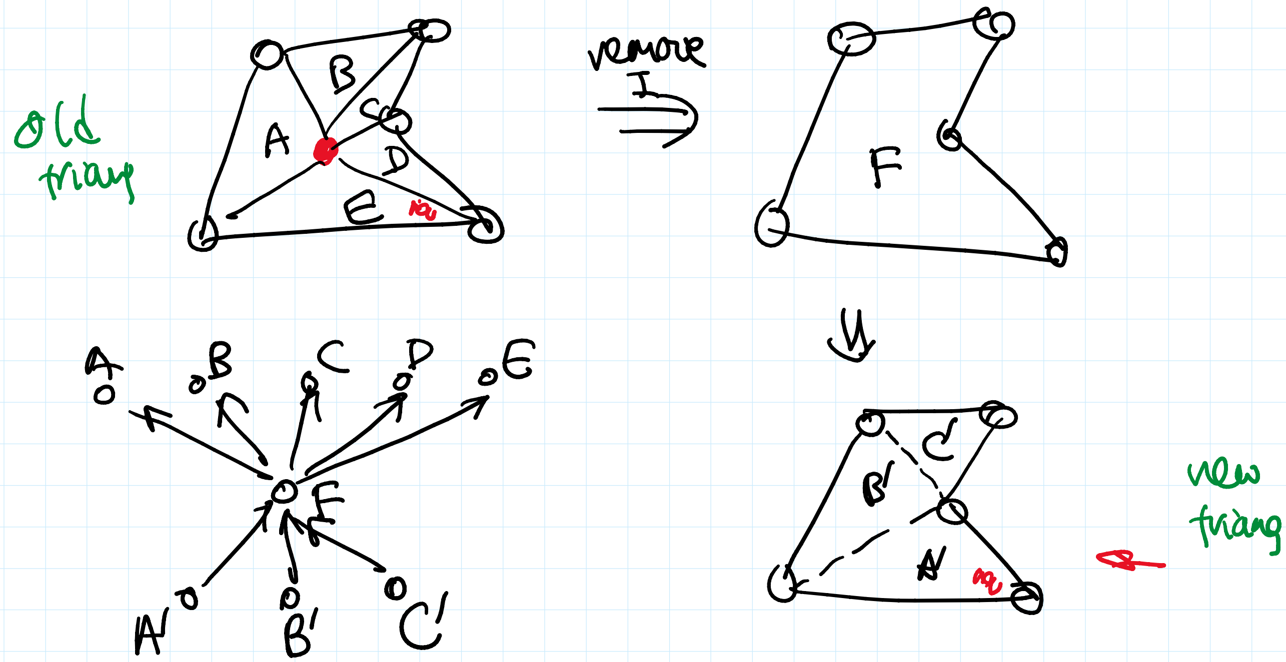
find a large independent set I of vertices (excluding 3 bdry vert.)

← no 2 vertices are adjacent

remove I from S

retriangulate, add ptrs between old & new triangles

}



not tree, but DAG

Query alg'm: just follow links in DAG

simple!

Fact Given any planar graph with n vertices, can find an indep set I of size $\geq \frac{n}{24}$ vertices in $O(n)$ time.

Furthermore, each $v \in I$ has deg ≤ 11.

Pf: Greedy alg'm:

repeat {
  pick some unmarked vertex v of deg ≤ 11
  add v to I
  mark v & its neighbors
}

Analysis:

at end, all unmarked vertices have deg ≥ 12

$$12\,(\#\text{unmarked vertices}) \leq \text{total deg} = 2m < 6n$$

$$\Rightarrow \quad \#\text{unmarked vertices} < \frac{n}{2}. \qquad (m < 3n)$$

each iteration marks ≤ 12 vertices

$$\Rightarrow \quad \frac{n}{2} < \#\text{marked vertices} \leq 12\,|I|$$

$$\Rightarrow \quad |I| \geq \frac{n}{24}. \qquad \square$$

Space $\quad S(n) \leq S\left(\frac{23n}{24}\right) + \underline{O(n)}$

vertices

$$\Rightarrow \quad O\left(n + \frac{23}{24}n + \left(\frac{23}{24}\right)^2 n + \cdots\right)$$

$$= O(24n) = \boxed{O(n)}$$

Preproc $\quad P(n) \leq P\left(\frac{23}{24}n\right) + O(n)$

Preproc time $P(n) \le P(\frac{23}{24} n) + O(n)$

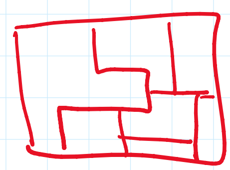
$\Rightarrow \boxed{O(n)}$

(init. triangulation step takes $O(n)$ time by chazelle '90)

Query time $Q(n) \le Q(\frac{23}{24} n) + O(1)$

$\Rightarrow O(\log_{\frac{24}{23}} n) = \boxed{O(\log n)}$.

Rmk - in word RAM model,



2D orthogonal point location

$O(n)$ space, $O(\log\log U)$ query time [C.'11]

for general case,

$O(n)$ space, $O\left(\frac{\log n}{\log\log n}\right)$ time [C.-Patrascu'06]

$O\left(\sqrt{\frac{\log U}{\log\log U}}\right)$

dynamic . . .

3D . . .

⋮

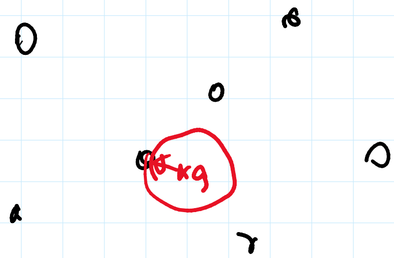
Nearest Neighbor Search in High Dims

Problem Given n pts P in $\mathbb{R}^d$,

build data structure s.t.

given query pt $q \in \mathbb{R}^d$, can find $p \in P$

minimizing $d(p,q)$.



Euclidean: $\sqrt{\sum_{i=1}^{d} (p_i - q_i)^2}$

L_1: $\sum_{i=1}^{d} |p_i - q_i|$

L_∞: $\max_{i=1}^{d} |p_i - q_i|$

Known: $d=2$: $O(n)$ space, $O(\log n)$ time by PL & Voronoi diag.

larger const d:

L_1 / L_∞: $O(n \log^d n)$ space, $O(\log^d n)$ query time

L_2: $O(n^{\lceil d/2 \rceil})$ space, $O(\log n)$ time

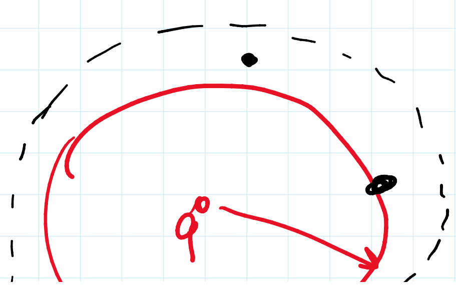
or $O(n)$ space, $\tilde{O}(n^{1 - \frac{1}{\lceil d/2 \rceil}})$ time

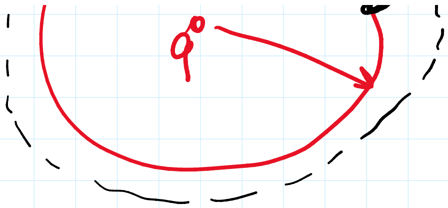
terrible for large d!

Relax problem: given $q \in \mathbb{R}^d$, find $p \in P$ s.t.

$$d(p,q) \leq c \cdot \min_{p' \in P} d(p',q)$$

approximation factor





Suffice to solve approx <u>decision</u> problem, for fixed radius r:

given $q \in \mathbb{R}^d$,

return some pt $p \in P$ with $d(p,q) \le cr$

or declare all pts $p \in P$ with $d(p,q) > r$.

(original prob can be solved by approx binary search
extra factor $\log U$)