Computing 2D Delaunay

Lower Bound $\Omega(n \log n)$

Many optimal algorithms

1) One approach
   a) Compute the Voronoi using sweep line $O(n \log n)$ BKOS Chap 7
   b) Dualize obtaining Delaunay $O(n)$

2) Divide-and-conquer

3) Random Incremental BKOS Chap 9
Random Incremental Delaunay

Input: \( P \subseteq \mathbb{R}^2 \) \( |P| = N \)
Big Triangle \( P_1, P_2, P_3 \) containing \( P \)

Simple Delaunay Tri \((P)\)

1) Make big \( \Delta \) containing \( P \)
2) Init: \( T \) to \( \Delta \) and assign \( P \) to \( \Delta \)
3) Randomly order \( P = \{ P_1, \ldots, P_n \} \)
4) For \( i = 1 \) to \( N \)
   a) Find tri \( t \in T \) containing \( P_i \)
   b) (Form cavity) Remove all tri \( t' \in T \) s.t. \( P_i \in C(t') \)
   c) (Form tent)
      \[ \forall e = (a, b) \text{ on } d \text{ Cavity form } \]
      \[ \text{tri } (P_i, a, b) \]
Delaunay Tri(\(P\))
Replace 4) with 4')

4') For \(i = 1\) to \(N\)

a) Find \(t \in T\) containing \(P_i\)
   i) If \(P_i\) in interior to \(t\) then split \(t\) into 3 Tri.
   ii) If \(P_i\) is on an edge between \(t\) & \(t'\)
       split into 4 Tri
d) While an edge opposite \(P_i\) is illegal.
   i) Flip the edge
e) Update data structure
Claim Delaunay Tri is correct

d.e. If step 4' starts with a Delaunay it returns a Delaunay.

The only illegal edges must "see" Pi

All new edges are Delaunay.

Timing

Lemma Expected \( \# \text{Tri created} \) is at most \( 9n+1 \)

\( V_r \equiv \# \text{tri at step } r \)

Backwards Analysis

if \( d \) = degree of removed vertex

\( V_r = 2d - 3 \quad \text{Note Euler } E(d) \leq 6 \)

\( E(V_r) < 2 \times 6 - 3 = 9 \)
Thm. Delaunay Tri is $O(n \log n)$ expected time & $O(n)$ expected storage.

Storage = # created tri = $O(n)$

Cost of point location?

$\triangle$ 2 lineside test per point in triangle

Flip = 1 lineside test per point in both triangles

Def. $C(\Delta) = \text{circum circle of } \Delta$

Note. Each flip starts with a new tri $\Delta$ & an old $\Delta'$.

Def. $K(\Delta) = \{ g : g \text{ interior } C(\Delta') \}$
\# lineside tests = \( O\left( \sum_{\Delta} \left| K(\Delta) \right| \right) \)

order cost by time tri is created.

Let \( P_r = P \cap P_r = r \cap \mathcal{C}_r = \text{Del} (P_r) \)

\# lineside tests = \( \sum_{r=1}^{n} \sum_{\Delta \in \mathcal{C}_r \setminus \mathcal{C}_{r-1}} \left| K(\Delta) \right| \)

\[ \text{Def} \ k(P_r, \delta) = \left| \{ \Delta \in \mathcal{C}_r : \delta \in C(\Delta) \} \right| \]

\[ K(P_r, \delta, \rho) \quad \text{" & \( P \) vertex \ D3"} \]
Let view cost as a table

\[
\begin{array}{ccc}
A_t & \mathcal{P}_t & \mathcal{P}_t(i) \\
\mathcal{T} & \bar{\mathcal{P}}' &=& \{\bar{P}'_1, \ldots, \bar{P}'_r\} \\
\text{Cost} &=& \sum_{i>\tau} \chi(\bar{P}'_r, \bar{P}'_j, \bar{P}'_r) \\
\end{array}
\]

Goal: Expected cost of a row

Instead: Expected cost of rth column & sum

**Def** Equivalence relation on \( S_n \)

\( \mathcal{T} \equiv \mathcal{T}' \) if \( \{T(1), \ldots, T(r)\} = \{T'(1), \ldots, T'(r)\} \)

We will uniformly bound each equivalence class.
Suffice to fix a set \( P_{r+1} \) and pick the last two elements from \( P_{r+1} \).

Claim: Row 8 sums to \( 3K(P_{r+1}/8, 8) \).

\[ \text{Det } \deg(8) = \text{degree of } 8 \text{ in } P_{r+1} \]

Thus \( K(P_{r+1}/8, 8) = \deg(8) - 2 \).

\[ \sum_{8} K(P_{r+1}/8, 8) \leq \sum_{8} (\deg(8) - 2) = \]

\[ = \sum \deg(8) - 2(r+1) = 6(r+1) - 2(r+1) = 4(r+1) \]
Sum over the Table

\[ \sum_k k(P_{m1} \mid 8, 8, p) \leq 3(4(r+1)) = 12(r+1) \]

\[
\text{Expected } (K(P_{m1}) \mid 8, 8, p) \leq \frac{12(r+1)}{(r+1)r} = \frac{12}{r}
\]

Back to the first Table

Expected value of a column \( r \leq \frac{12(n-r)}{r} \)

Expected row \( \leq \sum_{r=1}^{n} \frac{12(n-r)}{r} = 12nH_n - 12n = 12n(H_n - 1) \)