15-750
2/17/16
2/19/16

The Convex Hull Prob
(Sorting Prob of CG)

Def $A \subseteq R^d$ is convex if closed under convex combinations.

Def ConvexClosure$(A) \equiv CC(A)$ = smallest convex set $\supseteq A$

2 Defs of Convex Hull

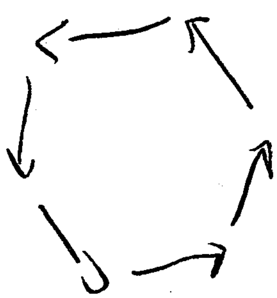
Def 1: $CH(A) = \partial CC(A)$ (Boundary)

Def 2: $CH(A) = CC(A)$

We will use Def 1

$A \equiv$ finite set

Thus in 2D $CH(A)$ is a simple closed polygon.
(say CCW)



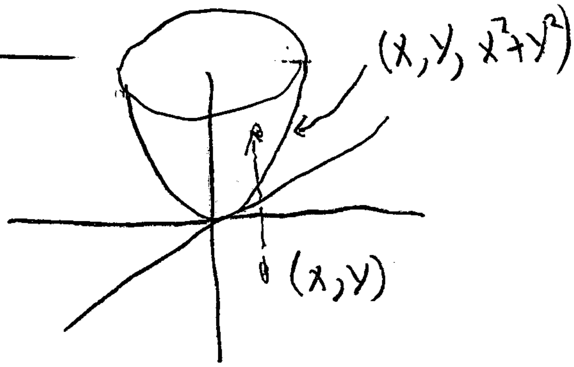
Lower bounds

Sorting reducible to C H

Input : $x_1, \cdots, x_n$

$CH\left((x_1, x_1^2), \cdots, (x_n, x_n^2) \right)$

The CH will be x_i's in sorted order.



An important use for CH

$P_1, \cdots, P_n \in \mathbb{R}^2$

$\bar{P}_i = (P_x, P_y, P_x^2 + P_y^2)$

$CH(\bar{P}_1, \cdots, \bar{P}_n)$ = Triangulated surface

The Delaunay Triangulation

We will use following characterization

<u>Claim</u> $[a,b]$ is on $CH(A)$ iff $\quad a \neq b$

1) $a, b \in A$

2) $\forall a' \in A$ either a' left of $[a,b]$

or $a' \in [a,b]$

2D Convex Hull by divide-and-conquer

$A = \{P_1, \ldots, P_n\}$ $\quad P_i = (x_i, y_i)$

Preprocess: sort A by x-coordinate

2D-CH(A)

<u>if</u> $|A| = 1$ return P_1

<u>else</u> $CH_L = $ 2D-CH$(P_1 \ldots P_{n/2})$

$CH_R = $ 2D-CH$(P_{n/2+1}, \ldots, P_n)$

STITCH(CH_L, CH_R)

STITCH(L, R)

Lowerbridge(L, R)

a = rightmost(L)

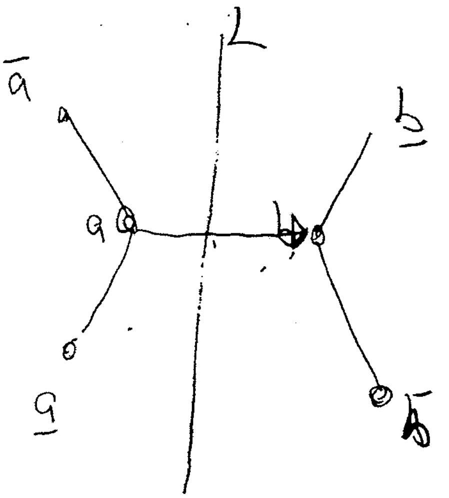
b = leftmost(R)

Repeat *) **)

*) While $\underline{a}$ Right(a,b) set $a \leftarrow \underline{a}$

**) While $\bar{b}$ Right(a,b) set $b \leftarrow \bar{b}$

Upperbridge(L, R) = ?



Correctness (termination)

*) generates triangles $(\underline{a}, b, a)$

**) " " $(a, \bar{b}, b)$

1) The Δ's are disjoint

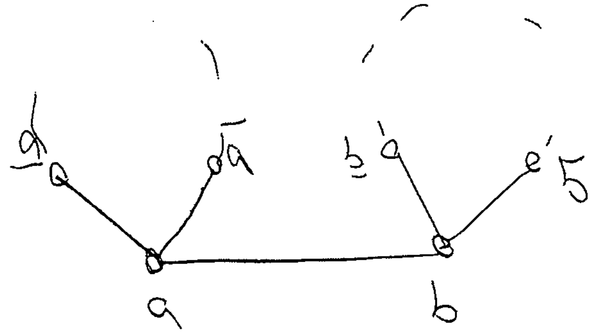
They are ordered by their intersection with vertical line L.

2) They are in CC(A).

Thus termination!

At termination $\underline{a}, \bar{a}, \underline{b}, \bar{b}$ are all left of (a,b)

ie



Since $(\underline{a}, a)$, $(a, \bar{a})$ and $(\underline{b}, b)$, $(b, \bar{b})$ are ~~on~~ $CH(L)$ & $CH(R)$ respectively.

Done

Timing: Preprocess $O(n \log n)$ to sort.

STITCH is $O(n)$

$$T(n) = 2T(n/2) + c \cdot n$$

$$\therefore T(n) = O(n \log n)$$

Random Incremental CH

Procedure RandomIncrementalCH(P)

0) Make $\Delta = (P_1, P_2, P_3)$ pick $c \in$ interior Δ

1) Construct ray from c to each P_i.

2) Partition P_i by edge of Δ they cross.

3) Randomly permute $P_4, \dots, P_n$.

   For i = 4 to n

   Let e be edge crossed by ray $c \to P_i$

   BuildTent(P, e)

Procedure BuildTent(P, e)

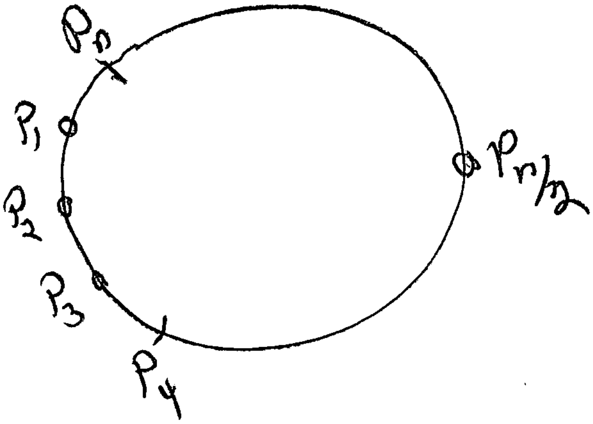
1) Find edges of CH "visible" to P by searching out from e.

2) Replace visible edges with 2 new edges

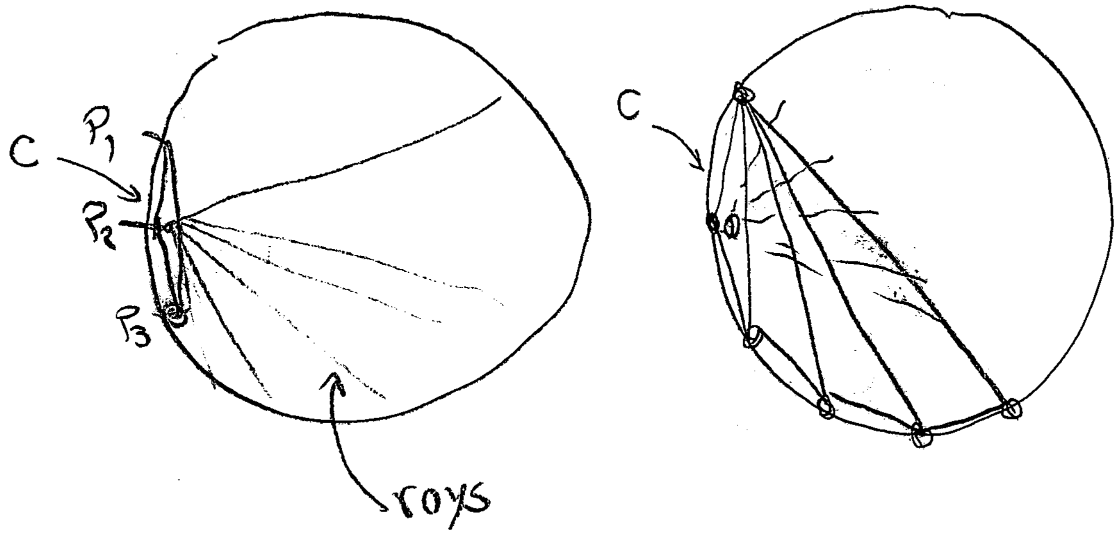
3) Assign rays to the new edges.

An Example

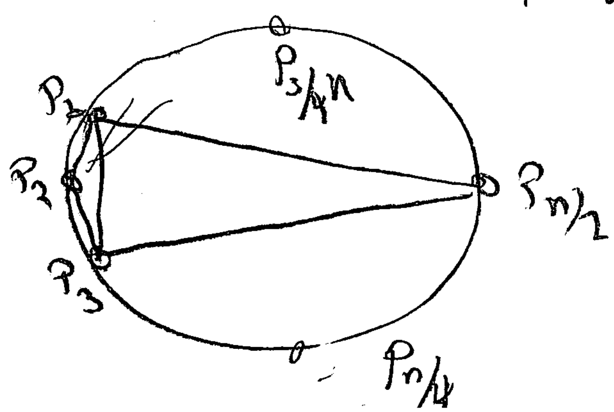
n-points on a circle



Worst case: Incremental order P_1 — P_n



"Best" Case $P_1 P_2 P_3$ $P_{n/2}$ $P_{n/4}$ $P_{\frac{3n}{4}}$ . . .



Correctness ?

Timing

Claim: $O(n)$ work other than BuildTent.

Consider steps 1 & 2 in BuildTent

1) at most $2n$ edges generated over life of alg.

2) Charging rule for line-side tests

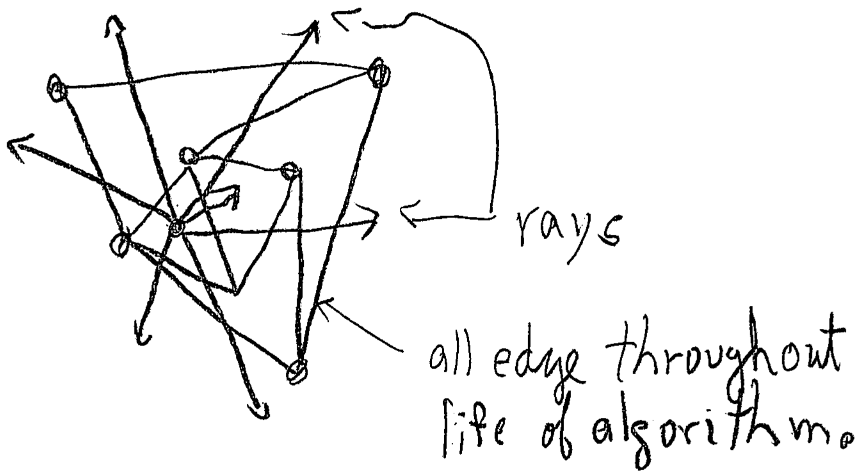
i) 2 not visible tests: we charge p_i

ii) each visible test: we charge to the edge

Total $2n + 2n$ or $4n$ tests.

Consider Step 3 in BuildTent

Cost to move point to new tent edge.



Claim upto multiplicative constant.

Step 3 $\approx$ #ray-edge crossings

$$V_i^j = \begin{cases} 1 & \text{if at time } i \text{ Ray}_j \text{ crosses removed edge.} \\ 0 & \text{o.w.} \end{cases}$$

Claim $\Pr[V_i^j = 1] \leq 2/i$

a) If Ray_j does not cross bdary edge at time i then

$V_i^j = 0$

b) $(v, w) = e$ bdry edge crossed by Ray_j.

Then V_i^j only if v or w picked.

Thus $\Pr[V_i^j = 0 \mid b)] = \frac{2}{i}$

Def $V_j = \sum V_i^j$

$E(V_j) = \sum E(V_i^j) \leq 2H_n$

Total expected cost $2nH_n$