The Convex Hull Prob
(Sorting Prob of CG)

<u>Def</u> $A \subseteq \mathbb{R}^d$ is <u>convex</u> if closed under convex combinations.

<u>Def</u> ConvexClosure$(A) \equiv CC(A) =$ smallest convex set $\supseteq A$

2 Defs of Convex Hull

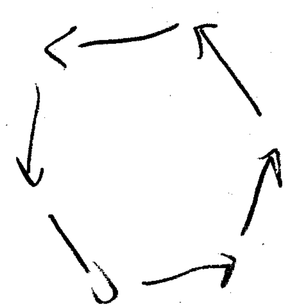
<u>Def 1</u> $CH(A) = \partial CC(A)$

<u>Def 2</u> $CH(A) = CC(A)$

We will use Def 1

$A \equiv$ finite set

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



15-456
1/25/13

The Convex Hull Prob
(Sorting Prob of CG)

<u>Def</u> $A \subseteq \mathbb{R}^d$ is <u>convex</u> if closed under convex combinations.

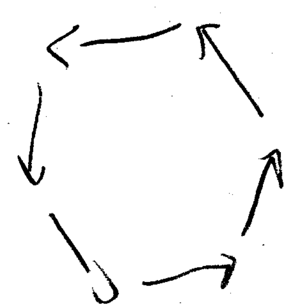
<u>Def</u> ConvexClosure$(A) \equiv CC(A)$ = smallest convex set $\supseteq A$

2 Defs of Convex Hull

<u>Def 1</u> $CH(A) = \partial CC(A)$

<u>Def 2</u> $CH(A) = CC(A)$

We will use Def 1
$A \equiv$ finite set
Thus in 2D $CH(A)$ is a simple closed polygon.
(say CCW)



We will use following characterization

Claim $[a, b]$ 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

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

Preprocess: sort A by x-coordinate

2D-CH(A)
  if $|A| = 1$ return P_1
  else $CH_L = $ 2D-CH$(P_1 \dots P_{n/2})$
    $CH_R = $ 2D-CH$(P_{n/2+1}, \dots, P_n)$
  STITCH(CH_L, CH_R)

STITCH(L, R)

Lowerbridge(L, R)

a = rightmost(L)

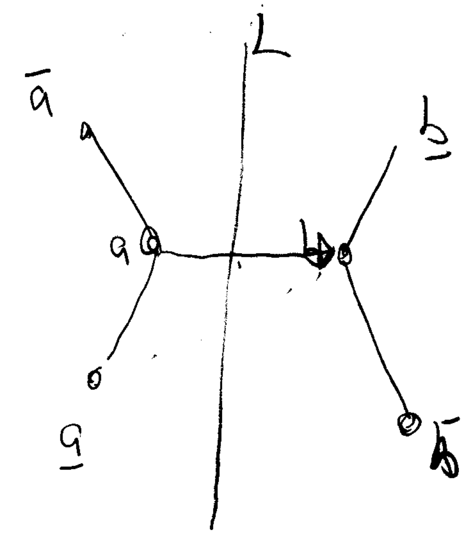
b = leftmost(R)

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

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

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

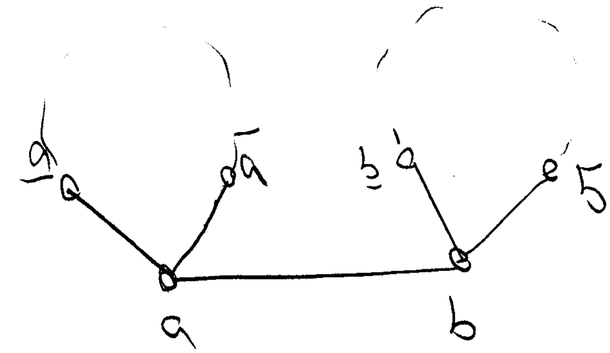
1) The Δ's are disjoint

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 $O(n)$

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

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

Lower bounds

Sorting reducible to C H

Input : $x_1, \dots, x_n$

$CH((x_1, x_1^2), \dots, (x_n, x_n^2))$

The CH will be x_i's in sorted order.

An important use for CH

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

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

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

The Delaunay Triangulation

Quick Sort & Backwards Analysis

Consider

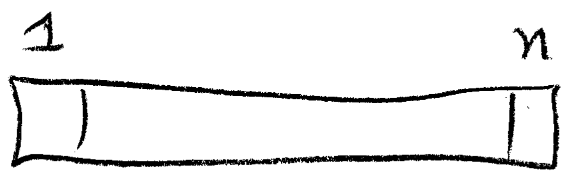
$QS(M)$ (distinct keys)

1) pick random $a \in M$
2) Split M: $S < a < L$ ($|M|-1$ comparisons)
3) return $QS(S) * a * QS(L)$

Goal: Expect # comparisons

Consider dart game:

Init: empty board

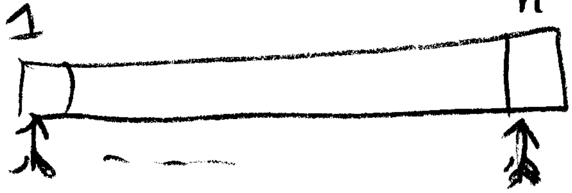


While nonempty square
  pick random empty sq
  cost = # empty sqs to left & right of dart.

Claim Expect Cost of dart game = Expect cost QS.

Backwards game:

Init: full board



While $\exists$ dart remove random dart.

Cost: # empty □s left & right.

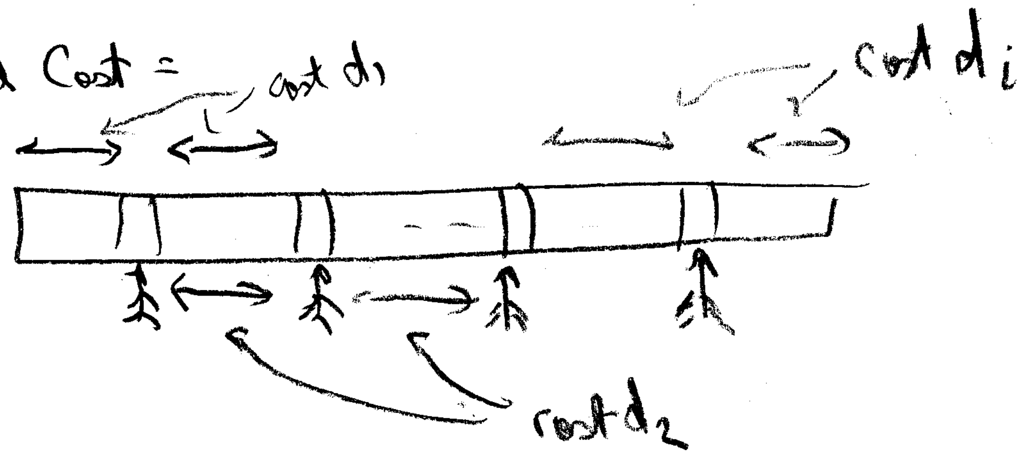
<u>Claim</u> Expect cost DG = Expect cost BW DG

Analysis backwards game

Asume i darts on board

T_i = Expected cost to remove random dart.

Total Cost =



$\leq 2(n-i)$

$$T_i \leq \frac{2(n-i)}{i} \leq \frac{2n}{i}$$

$$E(DG) = \sum T_i \leq \sum \frac{2n}{i} = 2n H_n = O(n \lg n)$$

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 $P_4, \ldots, P_n$.

   For $i = 4$ to n

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

   BuildTent(P, e)

Procedure BuildTent(P, e)

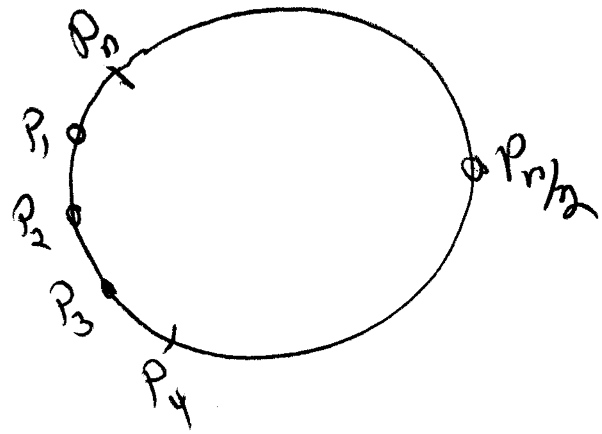
1) Find edges of CH "visible" to P by searching out from e.

2) Replace visible edges with 2 new edges

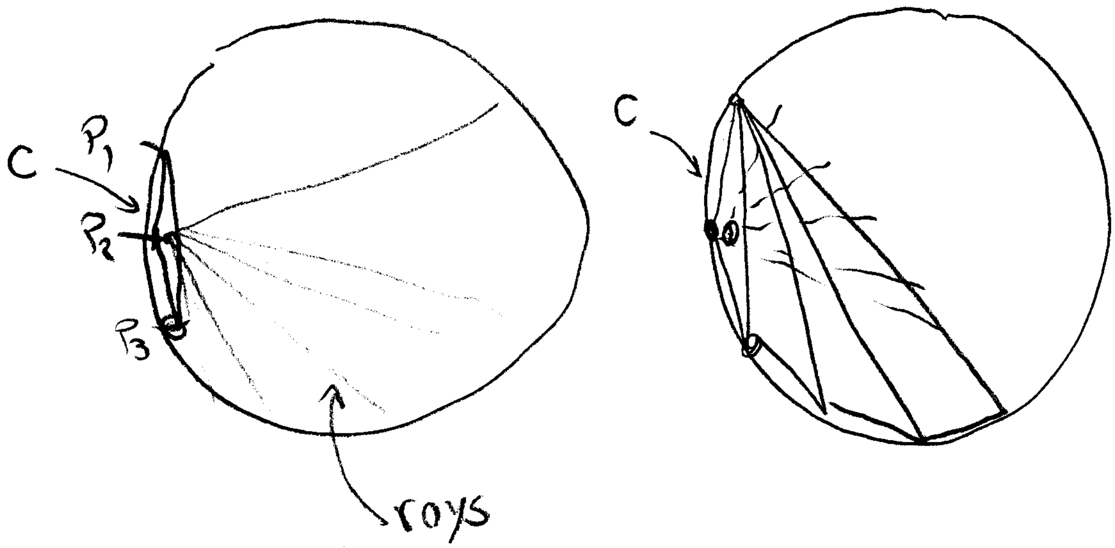
3) Assign rays to the new edges.

An Example

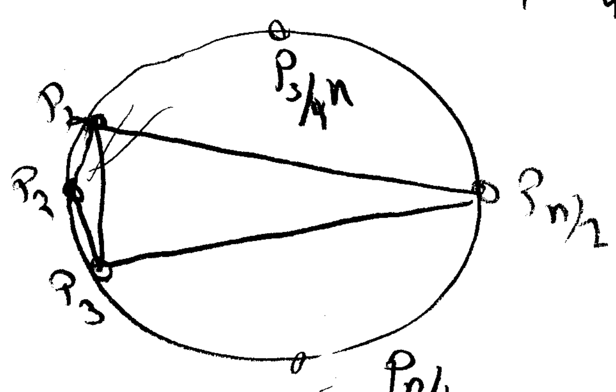
n-points on a circle



Worst case: Incremental ordr P_1 ~ P_n



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



Correctness ?

Timing

$O(n)$ 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

2 not visible tests: we charge P_i

each visible test: we charge to the edge

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

Consider step 3 in BuildTent.

Ray-costs

Backwards analysis

$i-3$ points to pick from say P_j

$$\mathrm{Cost}(P_j) = \begin{cases} 0 & \text{if } P_j \text{ not on hull} \\ \#\text{ray crossing to left \& right} & \text{o.w.} \end{cases}$$

C_i = cost

$$E(C_i) \le \frac{2(n-i)}{i-3}$$

C = total cost

$$E(C) = \sum_{i=4}^{n} E(C_i) \le \sum_{i=4}^{n} \frac{2(n-i)}{i-3} \le 2n \sum_{i=1}^{n} \frac{1}{i}$$