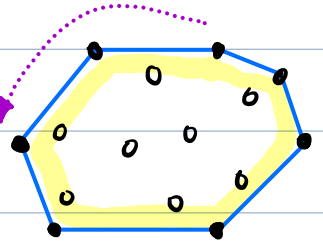
CMSC 754 - Computational Geometry
Lecture 3: Convex Hulls (Output sensitivity)

Recap:

- Given a point set $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^2$, compute $\text{conv}(P)$, the convex hull of P
- Output: Cyclic sequence of hull vertices
- Graham's Scan - $O(n \log n)$ algorithm

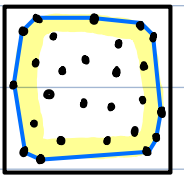


This lecture:

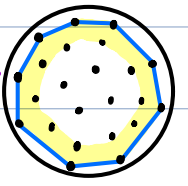
- Can we best $O(n \log n)$ time?
  - Jarvis's March $O(h \cdot n)$, h = no. hull vertices
  - $\Omega(n \log h)$ lower bound
  - Chan's algorithm - $O(n \log h)$ - optimal!

How many vertices do we expect on hull?

- n points uniform in square
  → Expect $O(\log n)$



- n points uniform in disk
  → Expect $O(n^{1/3})$

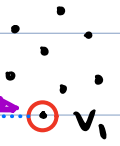


Jarvis's March - An $O(n \cdot h)$ algorithm

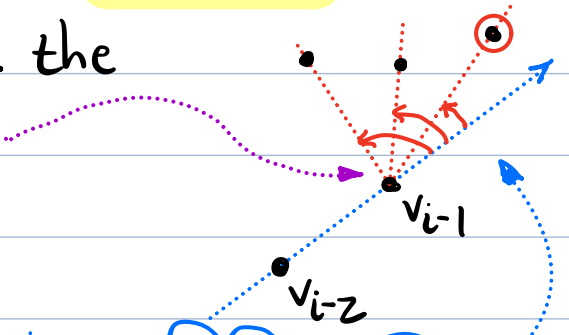
Idea:
- $v_1 \leftarrow$ any one vertex of hull
- for $i \leftarrow 2, 3, \ldots$
  - $v_i \leftarrow$ next vertex of hull
  - if $(v_i = v_1)$ return $\langle v_1, v_2, \ldots, v_{i-1} \rangle$

Details:

$v_1 \leftarrow$ point with min y-coord

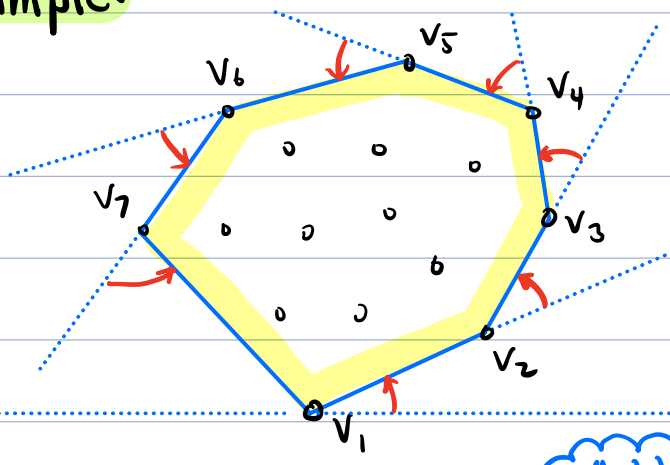


$v_i \leftarrow$ find the point that minimizes the turn angle w.r.t. the prior two vertices



Can compute pt. with min turn angle using orientation tests

Example:



Add sentinel point v_0 to get started

Correctness: (Exercise)

Running time:
- Compute $v_1 \rightarrow O(n)$
- Compute $v_i \rightarrow O(n)$

} Repeat h times

$O(h \cdot n)$ total

Lower Bound for Convex Hulls

CONV: Given a set P of n points in $\mathbb{R}^2$ compute the vertices of $\text{conv}(P)$ in cyclic order

> **Theorem**: In the algebraic decision-tree model, CONV has worst-case complexity of $\Omega(n \log n)$

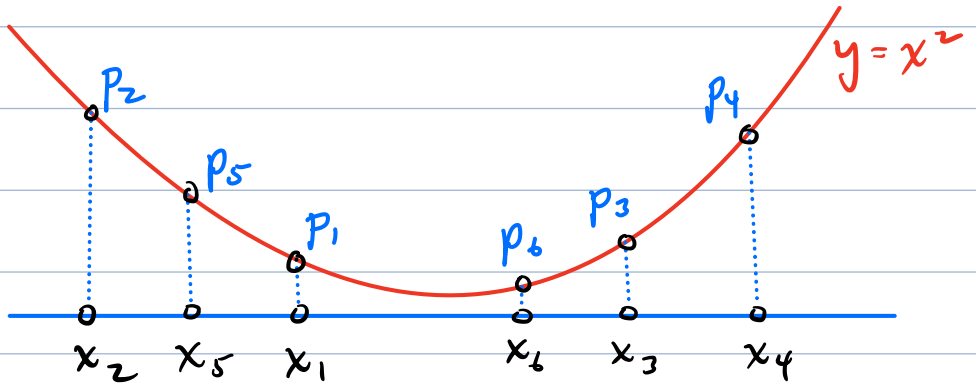
Algebraic Decision Tree Model: All the algorithm's decisions are based on signs of fixed-degree polynomials.

- Allows for orientation tests
- Disallows integer-based methods like { bucket sort, hashing }

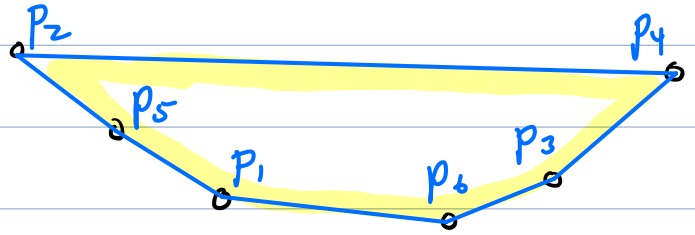
Proof: We will use the fact that sorting has a lower bound of $\Omega(n \log n)$ in the algebraic decision-tree model.

We'll reduce SORT to CONV in $O(n)$ time.
Given a set $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}$ to sort, generate point set $P = \{(x_1, x_1^2), \ldots, (x_n, x_n^2)\} \subseteq \mathbb{R}^2$

Reduction:
$O(n)$



Convex hull:
CONV



E.g. $\langle 6,3,4,2,5,1 \rangle$ (Cyclic order can start at any vertex)

Reorder:
$O(n)$

$\langle 2,5,1,6,3,4 \rangle$

By lower bound on sorting

Total time: $O(n) +$ CONV $+ O(n) \geq \Omega(n \log n)$

$\Rightarrow$ CONV $\geq \Omega(n \log n)$ □

This relies on fact that output of CONV is ordered.
If not? Lower bound still holds. (Harder proof!)

Theorem: Assuming algebraic decision-tree model, determining whether $\text{conv}(P)$ has h vertices requires $\Omega(n \log h)$ time.

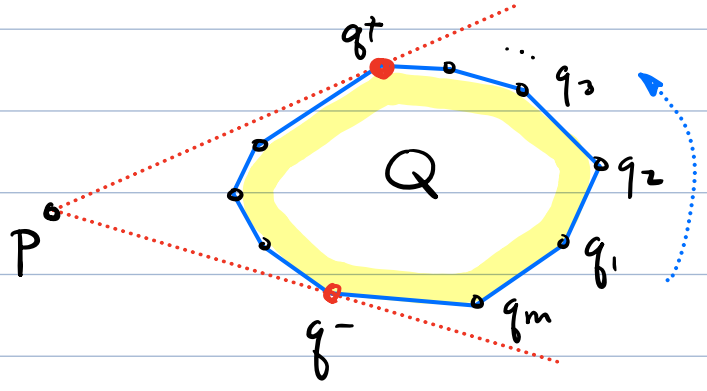
Proof: (see latex notes)

The ultimate algorithm for convex hulls - $O(n \log h)$
- Solved by Kirkpatrick + Seidel in 1986
- Quite complicated
- Chan's algorithm (1996) - Much simpler!
  - Clever combination of Graham + Jarvis

Chan's Algorithm - $O(n \log h)$ time
- Chicken + egg: Algorithm requires knowledge of h, but h is not known until end!

- Solves this by playing a clever guessing game.
- Utility lemma; will need later

Tangent Lemma: Given a convex polygon Q (as cyclic sequence of m vertices) and $p \notin Q$, can compute vertices q^+ + q^- s.t. $\overleftrightarrow{pq^+}$ + $\overleftrightarrow{pq^-}$ are supporting lines in $O(\log m)$ time

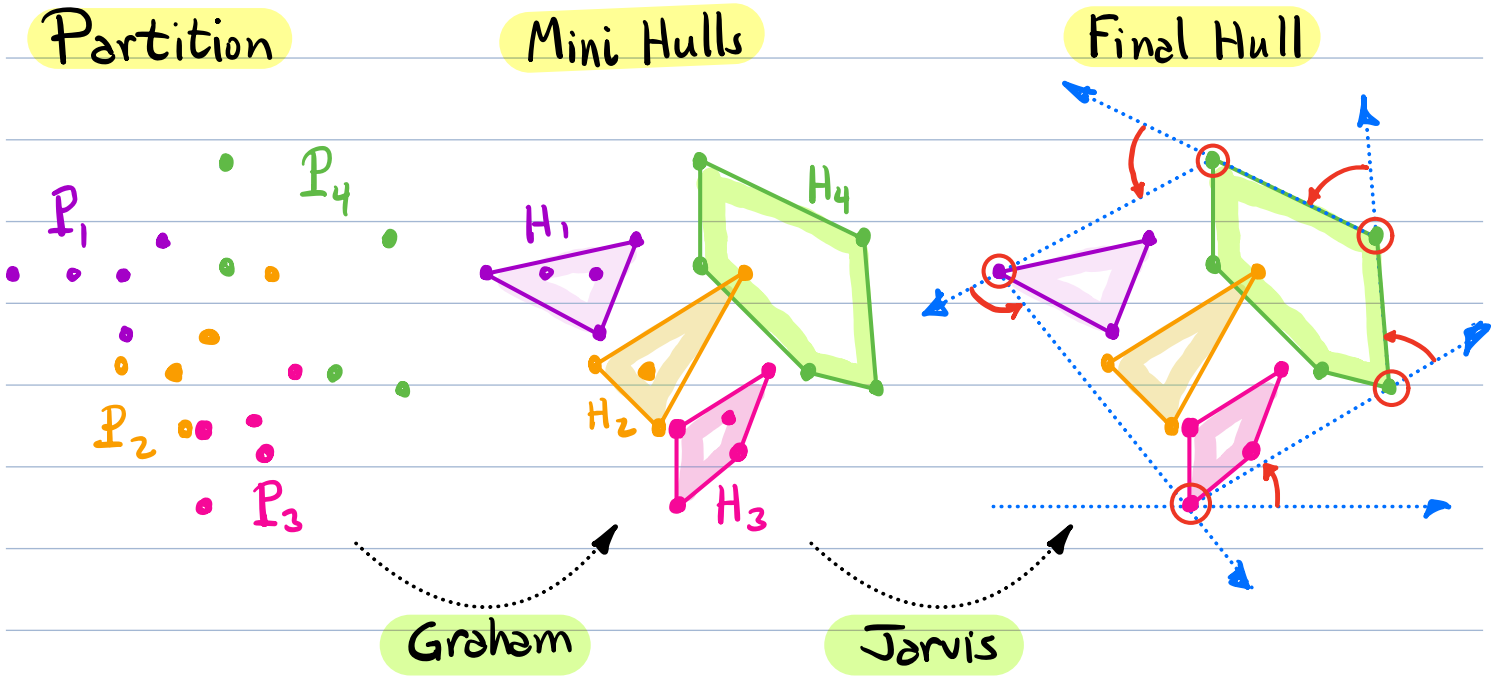


Proof: Exercise involving orientation tests + binary search. □

Elements of Chan's Algorithm:

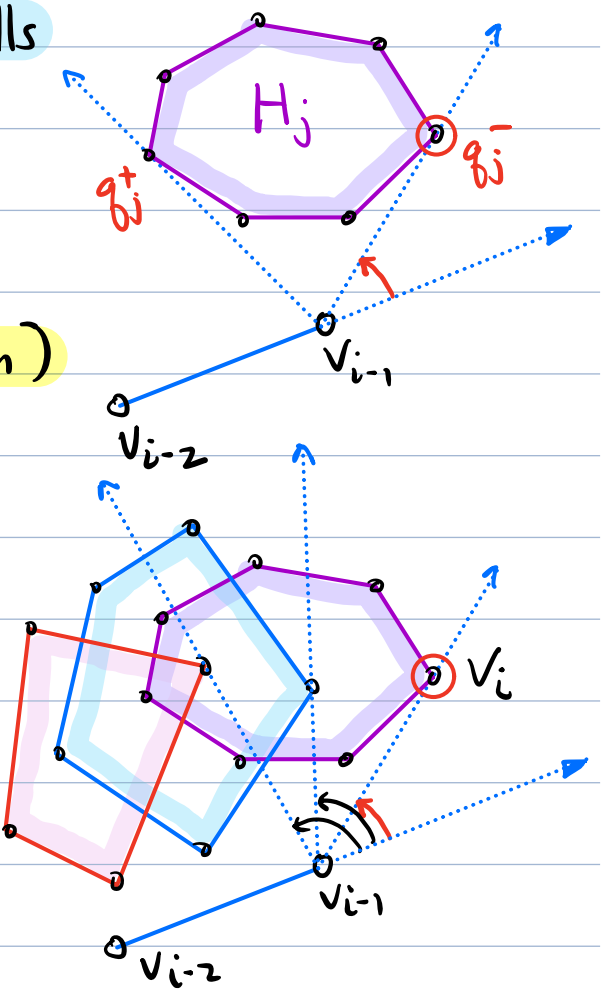
- Need to keep log factors to $\log h$
- Guess value of h - call it h^*
- Partition P into groups of size h^*
  - ⇒ Results in $k \approx n/h^*$ groups
- Run Graham's Scan on each group
  - Results in "mini hulls" $H_1, \ldots, H_k$
  - Time: $k \cdot O(h^* \log h^*) = (n/h^*)\, O(h^* \log h^*)$
    $= O(n \log h^*)$
  - Good, assuming $\log h^* = O(\log h)$
  - E.g. $h^* \le h^c$ (for constant c)
    - ⇒ $\log h^* \le \log h^c = c \cdot \log h = O(\log h)$ ☺
- Run Jarvis march on mini hulls
  - Treat each mini hull like a "fat point"
  - By Tangent Lemma, can compute each turn angle in time $O(\log h^*)$
  - Compute all turn angles in time $k \cdot O(\log h^*)$
  - Repeat for each hull vertex
    - ⇒ $O(h \cdot k \cdot \log h^*) = O(n(h/h^*) \log h^*)$
  - If $h \le h^* \le h^c$ ⇒ $O(n \log h)$ ☺

Example: $n = 20$, $h^* = 5$, $k = \lceil 20/5 \rceil = 4$

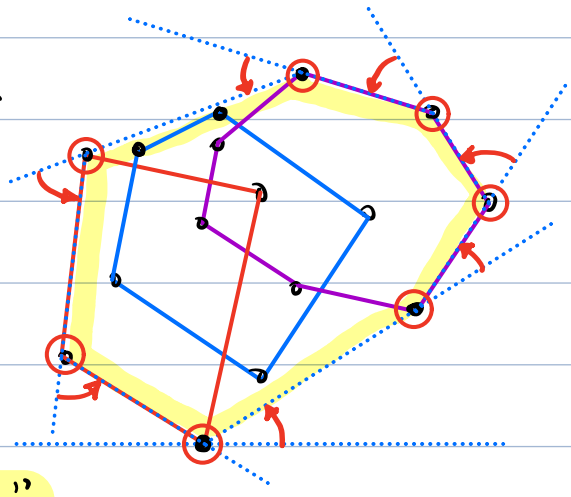


Details - Merging the Mini Hulls

- By the Tangent Lemma, we can compute both supporting lines in $O(\log m)$ time (we only need one)
- $m \le h^* \Rightarrow O(\log h^*)$
- compute tangent lines for all k mini hulls + take the smallest turn
- $O(k \cdot \log h^*)$
  $= O(\frac{n}{h^*} \log h^*)$



- Process terminates after h iterations



- Total: $O(h \cdot \frac{n}{h^*} \log h^*)$

- If our guess h^* is "close", $h \le h^* \le h^c$ (c = constant)

total time is:

Graham Phase: $O(k \cdot h^* \log h^*)$

$= O((n/h^*) h^* \log h^*)$

$= O(n \log h^*)$

$\le O(n \log h^c) = O(c \cdot n \cdot \log h)$

$= O(n \log h)$ ☺

Jarvis Phase: $O(h \cdot k \cdot \log h^*)$

$= O(h \cdot n/h^* \cdot \log h^*)$

$= O(n (h/h^*) \log h^*)$

$= O(n \log h^c)$ since $h \le h^* \le h^c$

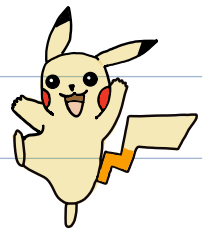
$= O(n \log h)$ ☺

Conditional Algorithm:

- Given our guess h^*, we run the above algorithm
- If we see $> h^*$ hull points in Jarvis phase — STOP + report failure (h^* too small)

How to play the Guessing Game?

- Overestimation is fatal, so start small + increase until success
- Arithmetic progression: $h_i^* = 2, 3, 4, 5, 6, \ldots, i$
  - Too slow $\Rightarrow O(n \cdot h \cdot \log h)$ [Exercise]
- Exponential progression: $h_i^* = 2, 4, 8, 16, \ldots, 2^i$
  - Better but still too slow
    $\Rightarrow O(n \log^2 h)$ [Exercise]
- Double exponential: $h_i^* = 2, 4, 16, 256, \ldots, 2^{2^i}$
  - Perfect!
    $\Rightarrow O(n \log h)$ [We'll prove this]



- You might think, any sufficiently fast series will work, but not true.
  Triple exponential: $h_i^* = 2^{2^{2^i}}$
  Too slow $\Rightarrow O(n \cdot h)$ [Exercise]

Final Algorithm:

Chan Hull (P)

```
h* ← 2
repeat
    h* ← (h*)^2        // h_i* = 2^(2^i)
    (status, V) ← conditionalHull(P, h*)
until (status = success)
return V
```

Correctness: Already explained

Time:

- Running time per iteration = $O(n \log h_i^*)$ where $h_i^* = 2^{2^i}$
- Stops (with success) when $h_i^* \geq h$

$$2^{2^i} \geq h \Rightarrow i \geq \lg \lg h$$

- Total time (up to constants)

$$\sum_{i=1}^{\lg \lg h} n \lg 2^{2^i} = n \cdot \sum_{i=1}^{\lg \lg h} 2^i$$

$$\leq 2 \cdot n \cdot 2^{\lg \lg h} \quad \text{(Geometric series)}$$

$$= 2n \lg h = O(n \log h)$$

Summary:

- Jarvis's March - $O(n \cdot h)$
- Computing convex hull requires $\Omega(n \log n)$ time (worst case)
- [Not proved] Counting no. of hull vertices requires $\Omega(n \log h)$ time
- Chan's algorithm
  - $O(n \log h)$ - optimal
  - Combination of Graham, Jarvis, + guessing game