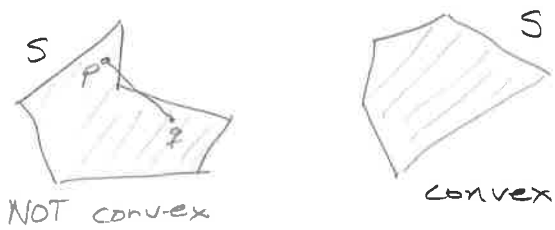
CPSC 516 Sept 20, 2022 (1)

Convex Hulls

A set S of points is convex if for every p and q in S, the points on the line segment $\overline{pq}$ are in S



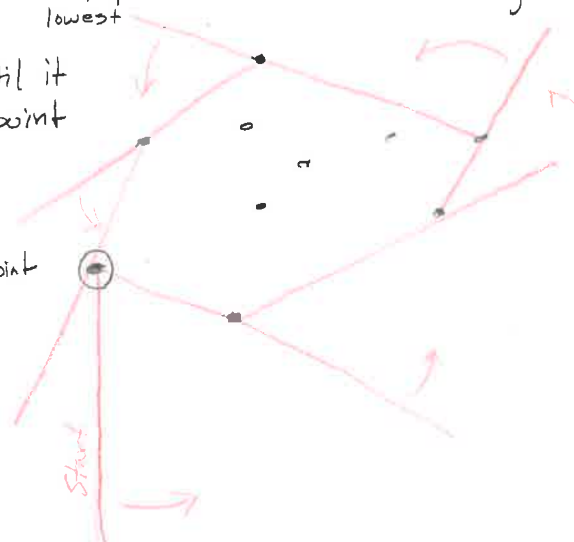
The Convex Hull of a set Q is the intersection of all convex sets that contain Q.

or its the "smallest" convex set that contains Q

Given n points in $\mathbb{R}^2$, find the convex hull.

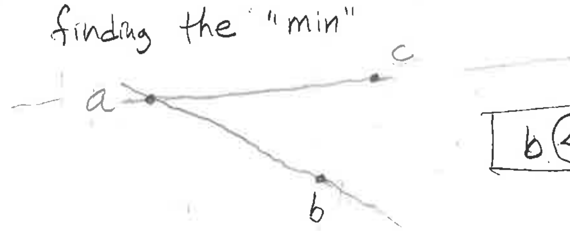
Gift wrapping [Jarvis 73]

- Tie a string to leftmost, lowest point (and let it hang down)
- Swing the string ccw until it first touches another point
- Tie it to that point
- Repeat until back to first point



We can't really "swing the string"

But we can find the next tie point by finding the "min"



$\boxed{b < c}$ because c is "left" of the line ab

$$\text{Area2}(a\ b\ c) = a_x b_y - a_y b_x + a_y c_x - a_x c_y + b_x c_y - c_x b_y > 0$$

$\iff$ a to b to c is Left turn

CPSC 516 20 Sept 2022 (2)

Gift Wrapping running time: $O(n \cdot h)$ h = # hull vertices

So if you think the hull is small, this is a good alg. to try.

$\boxed{\text{But}}$ h might be $\Omega(n)$ $\Rightarrow$ worst case $\Omega(n^2)$ (as function of n)

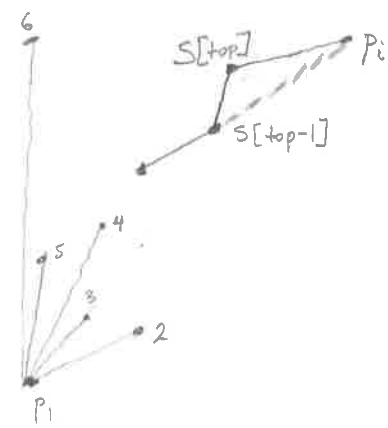
Graham's Scan 1972 (before Jarvis – the only reference in Jarvis)

1. Find point p_1 in Q with smallest y-coord (break ties by smallest x-coord)
2. Sort remaining points in Q ccw around p_1. Let $p_2\, p_3 \cdots p_n$ be these points in order of increasing slope from p_1
3. Start with $p_1\, p_2\, p_3$ as CH of first 3 points and put them on stack S

4. For $i = 4$ to n
   - while noleftturn from $S[top-1]$ to $S[top]$ to p_i
     pop S
   - push p_i onto S

Is p_2 ever popped?

5. Return S



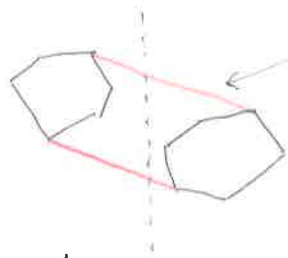
Running time $\Theta(n \log n)$

Pushing and Popping take linear time

Sort takes $\Theta(n \log n)$

Other $\Theta(n \log n)$ algorithms

- Incremental (add points to hull in order of x-coord)
- Divide and Conquer (• split point set in half using a vertical line
  • recursively solve both halves
  • add upper and lower "bridges")

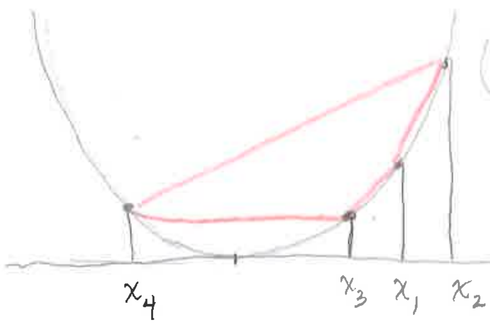


Lower bound

- How to Sort using Convex Hull algorithm

1. Given $x_1\ x_2 \dots x_n$ (numbers to sort)
2. Project them onto a parabola — $O(n)$ time

$$(x_1, x_1^2)\quad (x_2\ x_2^2)\ \dots\ (x_n, x_n^2)$$



3. Give these point as input to any $\boxed{\text{CH algorithm}}$ A — $T_A(n)$
4. Starting at the leftmost point, walk CCW around the hull and output the x-coord of each hull vertex. — $O(n)$ time

Total time $= T_A(n) + O(n)$

Since $\boxed{\text{all sorting algs take time } \Omega(n \log n)}$

$$T_A(n) + O(n) \in \Omega(n \log n) \Rightarrow T_A(n) \in \Omega(n \log n)$$

But we know this for comparison-based models of computation

Our model of computation for CH must be stronger to even hope to solve CH.