Convex hull
Convex

A convex set means any line that starts and ends within the set does not leave the set.

not convex, outside

convex
The **convex hull** is a convex set with the smallest area

Today we want to find the convex hull of a set of points $p_0, p_1, \ldots, p_m$

Give the set of points on the convex hull (in counter clockwise order)
Jarvis's March

The first method is like wrapping a gift (or putting a string around pins)
Jarvis's March

First start at one extreme side, we choose minimum y value, call it $p_0$

Then find point $p_1$ that minimizes the angle between $p_0$ and $x+$ axis
Jarvis's March

Continue this process until you reach the highest point (largest y value)

The find the smallest angle between the x-axis and your current point
Jarvis's March

Jarvis's March($p_0, p_1, \ldots, p_m$)

current = argmin $p.y$, hull = empty

while(current $\neq$ argmax $p.y$)
    hull = hull U current
    current = smallestAngle(current, x+)

while(current $\neq$ argmin $p.y$)
    hull = hull U current
    current = smallestAngle(current, x-)
Jarvis's March

\[ \text{argmax} \ p.y \]

\[ \text{argmin} \ p.y = p_0 \]

find min angle
Jarvis's March

Run time? (what is worst case?)
Jarvis's March

Run time?

Both while loops run total: $h$ times, where $h$ is number of points on hull.

Each loop needs to find minimum angle, which takes $O(m)$ time.

$O(hm)$, worst case $O(m^2)$ (circles)
Jarvis's March

Instead of needing to compare to either the $x^+$ or $x^-$ axis...

... could compare from previous point (Don't as takes more computation)
Graham's scan

Another method is to keep a working copy of the convex hull and add in one point at a time.

Again, we will start with argmin $p.y$

This time we simply add every point in counter clock-wise order.
Graham's scan

To be in the convex hull, and go counter clock-wise you cannot make a right turn (non-left)

We simply add each point and check to see if we turned right

If we did, we can discard this point
Graham's scan

When a remove a node, we backtrack to see if more need to be removed

(left/right = use cross product)
Graham's scan

Graham-Scan\((p_0, p_1, \ldots, p_m)\)

Let \(p_0 = \text{argmin } p.y\)

Sort \(p_i\) in smallest angle polar coord

if \(m < 3\) then no hull, quit

push \(p_0, p_1, p_2\) into \(S\)

for \(i = 3\) to \(m\)

    while angle under-top\((S)\), top\((s)\), \(p_i\) is nonleft turn

        pop\((S)\)

        push \(p_i\) into \(S\)

return \(S\)
Graham's scan

https://www.youtube.com/watch?v=dw120YcIav8
Graham's scan

Run time:
Sorting takes $O(m \lg m)$
for loop runs $O(m)$
All pushes happen once per point,
push constant time, so $O(m)$
Similarly pops $O(m)$

Total: $O(m \lg m)$
Graham's scan

Correctness: induction
Hypothesis = S contains convex hull for up to point $p_i$ at start of for loop

Base: $p_0$, $p_1$ and $p_2$ in S, makes a triangle which is convex

Termination: runs through all points
Graham's scan

Step: start of for, know that convex hull for i-1

After popping until next node is $p_j$, we know this node convex hull for first j points

Show adding i to $p_j$ encloses points