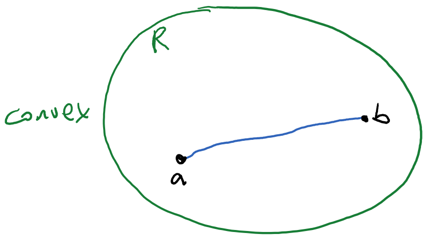
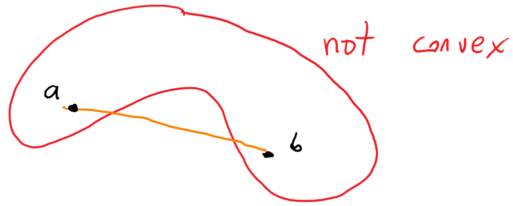
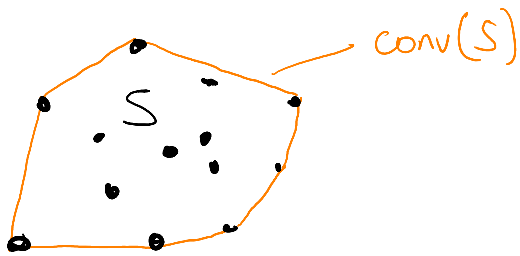
A region R is **CONVEX** if, for any two points a and b in R, the line segment $\overline{ab}$ is also in R.



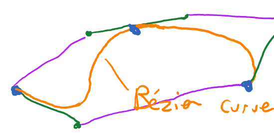


Let S be a set of points. The **CONVEX HULL** of S, denoted conv(S), is the intersection of all convex regions containing S.



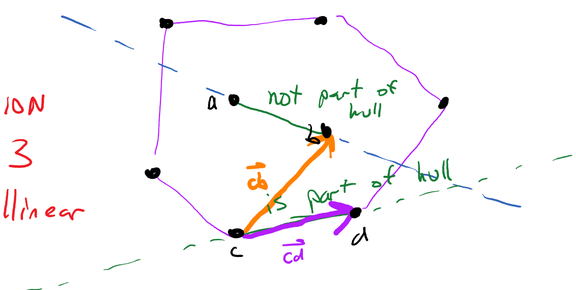
Applications:

collision detection
geographic information systems
optimization
geometric modeling



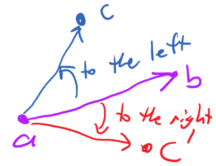
QUESTION: Given coordinates of points in the plane, how would you program a computer to find their convex hull?

GENERAL POSITION
assume no 3 points are collinear



Points a and b are consecutive hull vertices if and only if all other poins of S are on one side of the line through a and b.

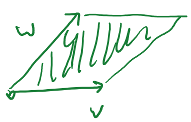
Question: Given points a, b, c, determine whether c is to the left of $\overrightarrow{ab}$.

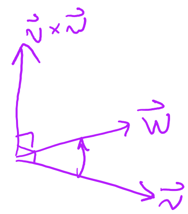


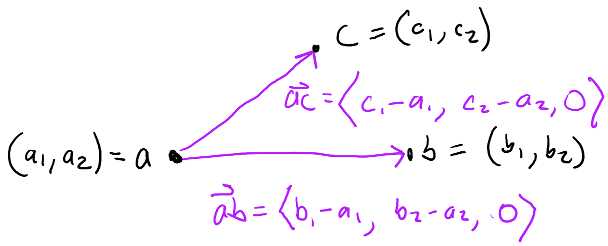
idea: absolute value of the cross product $\vec{v} \times \vec{w}$ gives the signed area of a parallelogram

$+$ if $\vec{w}$ is left of $\vec{v}$,

$-$ if $\vec{w}$ is right of $\vec{v}$







"left-of" query

Cross product: $\overrightarrow{ab} \times \overrightarrow{ac} = \langle b_1 - a_1, b_2 - a_2, 0 \rangle \times \langle c_1 - a_1, c_2 - a_2, 0 \rangle$

$$= \langle 0, 0, \underbrace{(b_1 - a_1)(c_2 - a_2) - (b_2 - a_2)(c_1 - a_1)}_{\text{call this } Q} \rangle$$

If $Q > 0$, then c is to the left of $\overrightarrow{ab}$.

If $Q < 0$, then c is to the right of $\overrightarrow{ab}$.

If $Q = 0$, then c is collinear to $\overrightarrow{ab}$.

Naïve Algorithm: For each pair of points a, b in S, if all other points in S are left of $\overrightarrow{ab}$, then $\overrightarrow{ab}$ is a hull edge.

pseudocode:

```
hull = {}
for each a in S:
    for each b≠a in S:
        for each c≠a,b in S:
            if c is not left of ab,
                then ab is not a hull edge, so continue
        append ab to hull
output hull
```

problem: If there are n points, then this algorithm requires $O(n^3)$ operations, which is inefficient.

INCREMENTAL ALGORITHM

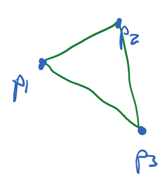
INPUT: set S of n points

OUTPUT: a list L of vertices of $\text{conv}(S)$, in counterclockwise order

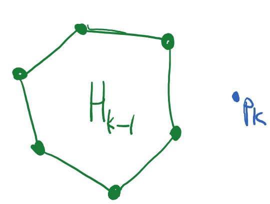
ALG:

1. Sort the points by x-coord. $p_1, p_2, p_3, \dots, p_n$
   ↳ $O(n \cdot \log n)$

2. Let H_3 be a list containg p_1, p_2, p_3 in counterclockwise order.



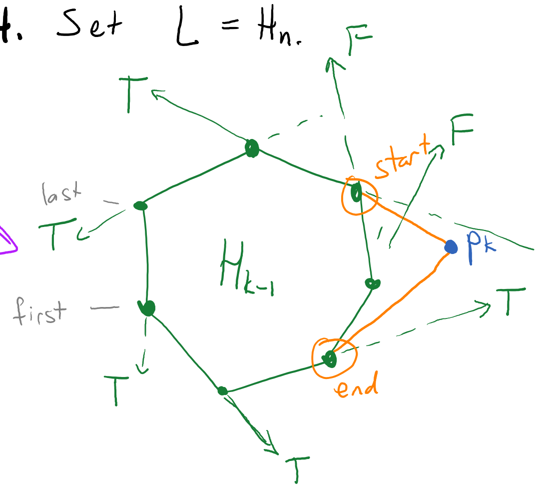
3. For k from 4 to n:
   Consider H_{k-1} with p_k.



How to do this?

Remove any interior points, insert p_k to form a list H_k, which is the convex hull of $p_1, p_2, \dots, p_k$ in counterclockwise order.

4. Set $L = H_n$.



subroutine "find visible":

input: hull H_{k-1}, point p_k

output: end and start indexes

algorithm:

```
prev = leftOf(last(H_{k-1}), first(H_{k-1}), p_k)
for i = 1 to k-2:
    next = leftOf(H_{k-1}[i], H_{k-1}[i+1], p_k)
    if prev and not next:
        set end = i
    if (not prev) and next:
        set start = i
return (end, start)
```

Form H_k like this:

p_k — add this vertex

$[p_{i_1}, p_{i_2}, p_{i_3}, \cancel{p_{i_4}, p_{i_5}}, p_{i_6}, p_{i_7}, p_{i_8}]$

end index

remove these vertices

start index