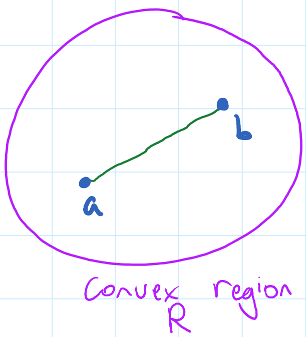
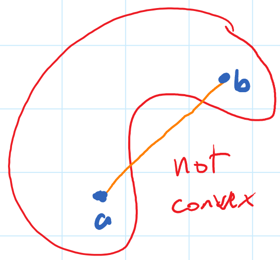
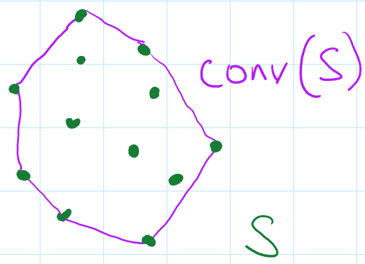
A region R is **CONVEX** if, for any two points a and b in R, the line segment ab is in R.

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

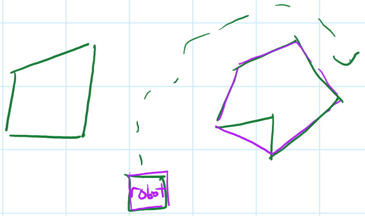






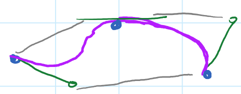
Application of Convex Hulls

- Collision detection (e.g. robot motion planning)



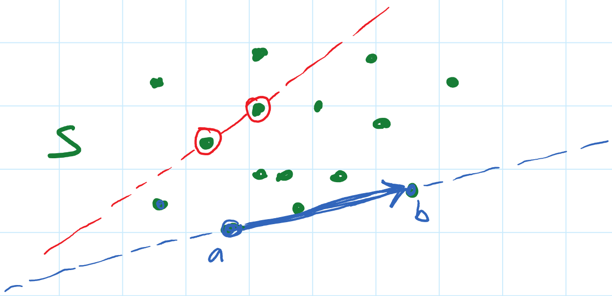
- geographic information systems
- optimization problems (e.g. simplex algorithm)
- geometric modeling

  e.g. a Bézier curve lies within the convex hull of its control points



Question: Given coordinates of a set of points S in the plane, how would you program a computer to find $\text{conv}(S)$?

IDEA: Points a and b are ~~hull~~ consecutive hull vertices if and only if all other points of S lie on the same side of edge $\overline{ab}$.



ALGORITHM:

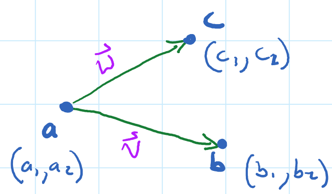
let n = number of points

```
hull = {}
for point a in S:
    for point b ≠ a in S:
        if all points c ≠ a,b in S are left of ab⃗:
            append edge ab to hull
return hull
```

3 nested loops over all points: $\boxed{O(n^3)}$

How to determine "left of"?

idea:

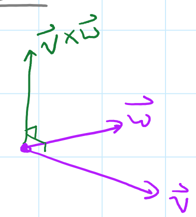


Is c left of $\vec{ab}$?

$\vec{w} = \langle c_1 - a_1,\ c_2 - a_2,\ 0 \rangle$

$\vec{v} = \langle b_1 - a_1,\ b_2 - a_2,\ 0 \rangle$

3D

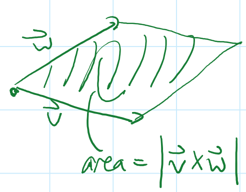


cross product: $\vec{v} \times \vec{w} = \langle 0, 0, \underbrace{(b_1 - a_1)(c_2 - a_2) - (b_2 - a_2)(c_1 - a_1)}_{\text{let this be } Q} \rangle$

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

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

If $Q = 0$, then c is collinear with $\vec{ab}$.



Algorithmic complexity:

"big O of n^3"

We say an algorithm is $\boxed{O(n^3)}$ if its runtime is not greater than $c \cdot n^3$ for some constant c (and large n).

Question: Can we do better?
Can we compute $\mathrm{conv}(S)$ more efficiently?