Given points in d-dimensional space, find a good “container” = convex polytope. Many applications, e.g. collision detection, pattern recognition, motion planning . . .

In 2D, imagine putting a rubber band around the points. In 3D, wrap with shrink-wrap.

More formally:

A set is *convex* if for every two points \( p, q \) in the set, all points on the line segment \( pq \) are also in the set.

The *convex hull* of set \( S \) is the intersection of all convex sets that contain \( S \).

Note that the convex hull of \( S \) is convex. The fact that the convex hull of a set of points \( S \) is a convex polytope whose vertices are points of \( S \) requires a proof, which we will do later.
Convex Hull Algorithms in 2D

Almost any algorithmic paradigm will work, so this problem is a great one for Algorithms courses.

**Incremental Algorithm** — add points one by one in sorted order by x coordinate
Graham’s Algorithm

Another sorting-base approach.
Sort the points radially around some point X inside the convex hull.
Divide and Conquer Algorithm

Divide the points in two by a vertical line (easy if we sort by x coordinate). Recurse on each side. Then combine the two sides.
Lower Bound

There is an Omega(n log n) lower bound on computing the ordered convex hull in 2D on a RAM with +,-,x.
Output sensitive algorithm

Idea:

Express the run time as a function of input size, \( n \), and output size, \( h \).

Gift-Wrapping (or “Jarvis’s March”)
So, what is the best algorithm in terms of $n$ and $h$?

$O(n \log h)$ algorithm — first developed by Kirkpatrick and Seidel 1986, uses linear time median finding.

improved by Timothy Chan, 1996.

**Chan’s Algorithm**
Next: convex hull in higher dimensions.