6.Computational Geometry

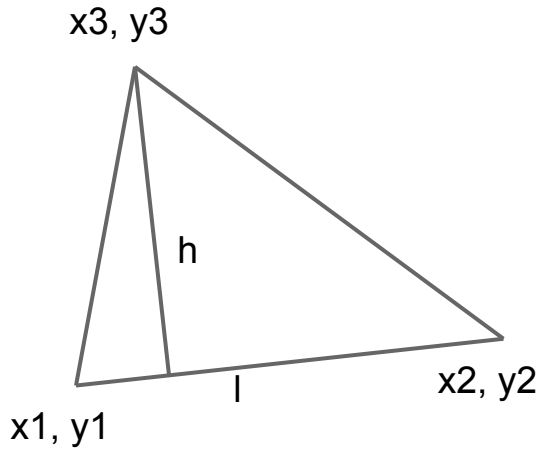
Computational Geometry

1. Useful for imaging, games etc.
2. Fundamental algorithms
3. Often solution ideas are simple
4. Not easy to implement
    a. arithmetic errors
    b. many awkward edge cases

Polygon Area

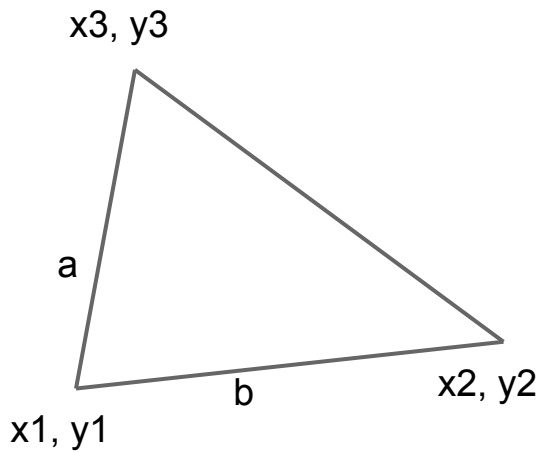
Given a set of points representing the vertices of a polygon, find its area.

Polygon Area: Triangle



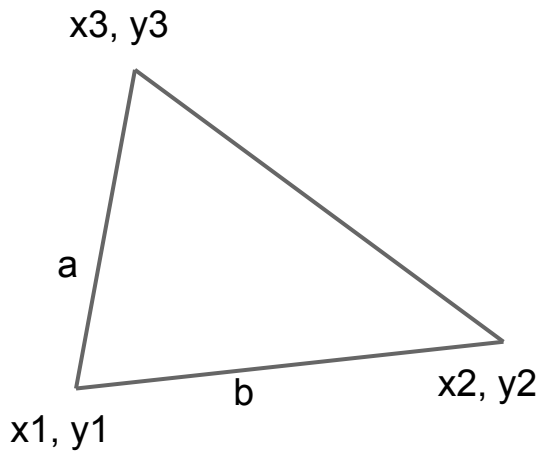
1. Area = (h x l) / 2

Polygon Area: Triangle



1. Area = (h x l) / 2
2. Area = a x b x sin(a, b)

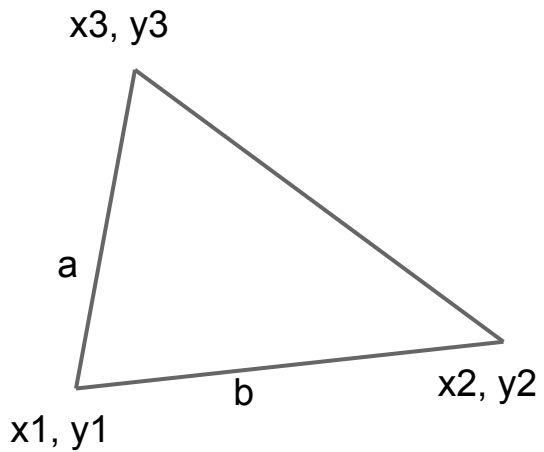
Polygon Area: Triangle



1. Area = (h x l) / 2
2. Area = a x b x sin(a, b)
3. Area = .5 * det(x1 y1 1
   x2 y2 1
   x3 y3

1)

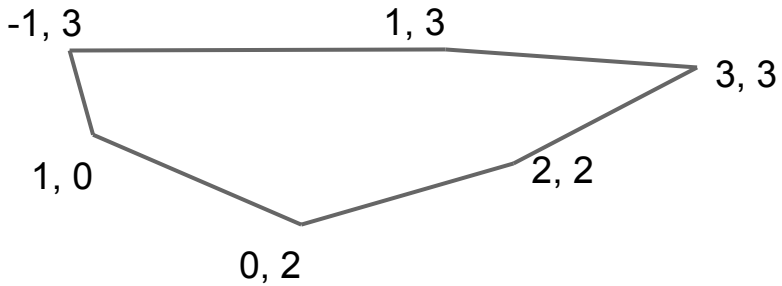
Polygon Area: Triangle



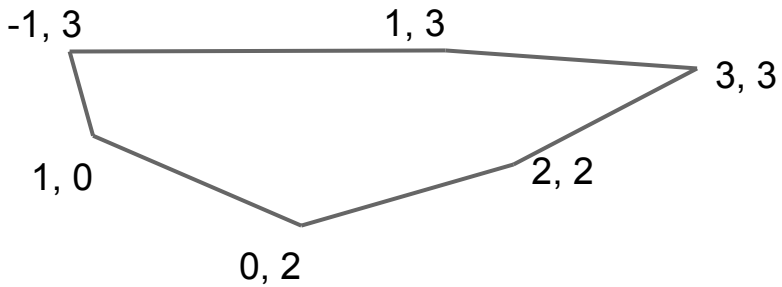
1. Area = (h x l) / 2
2. Area = a x b x sin(a, b)
3. **Area = .5 * det(x1 y1 1**
   x2 y2 1
   x3 y3 1)

Easiest to use when given coordinates

Polygon Area

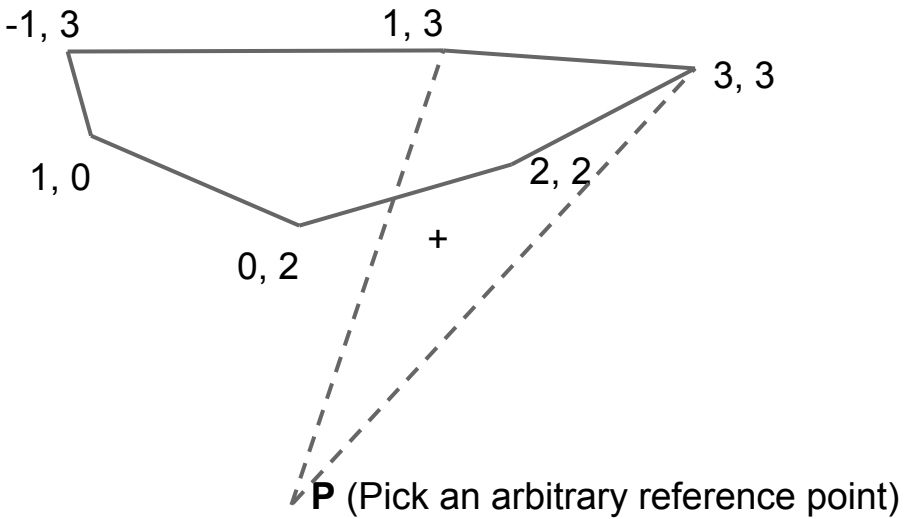


Polygon Area

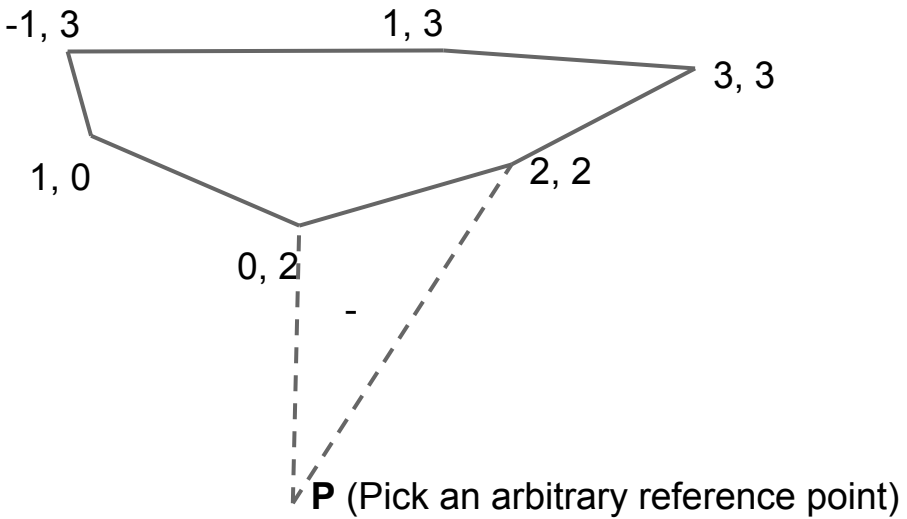


P (Pick an arbitrary reference point)

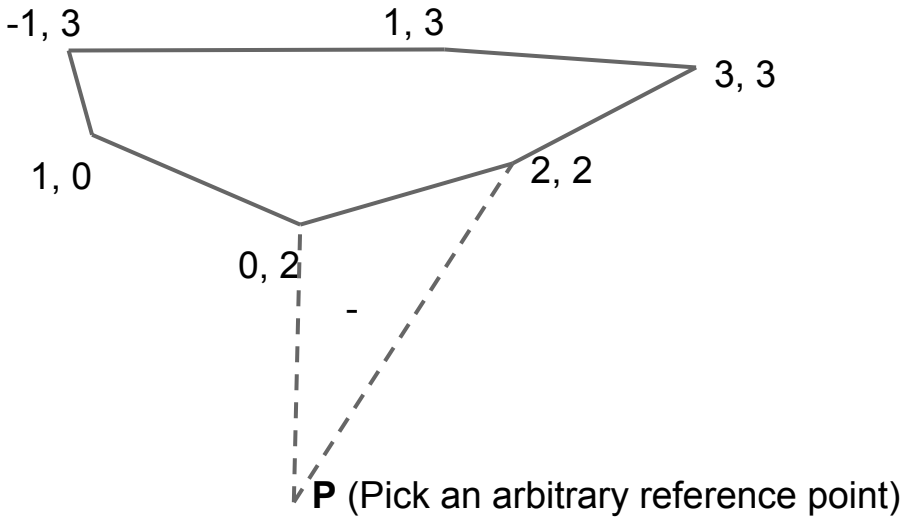
Polygon Area



Polygon Area



Polygon Area



1. Pick reference Point (P = 0, 0)
2. Traverse points trigonometrically
3. Sum up signed areas
    a. A += x1*y2 - x2*y1;
4. Done!

Polygon Area

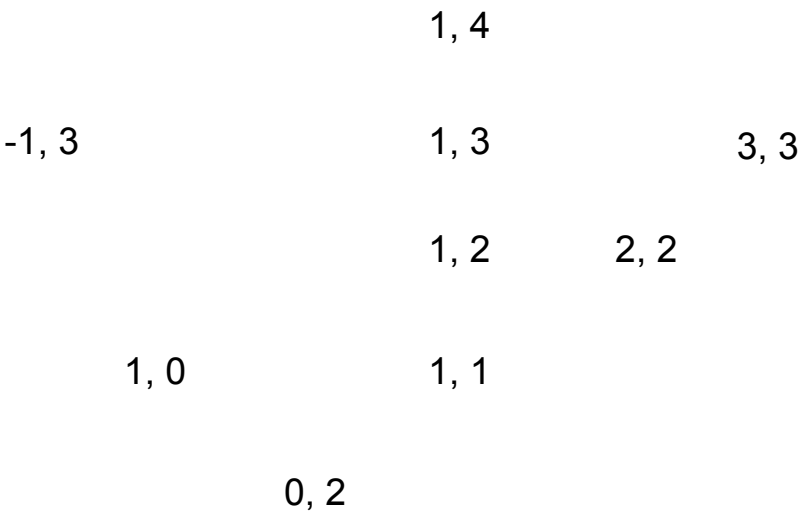
1. Keep it Simple!
   a. Use determinant formula

2. Double precision may not be enough
   a. Use long double or BigDecimal

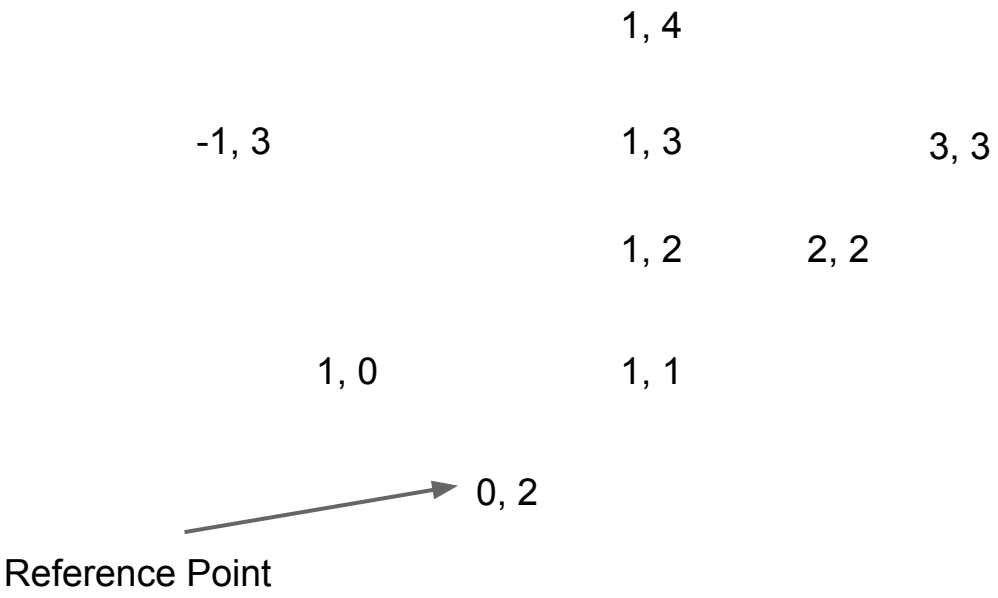
Convex Hull

Given a set of points find the smallest area polygon that contains all points.

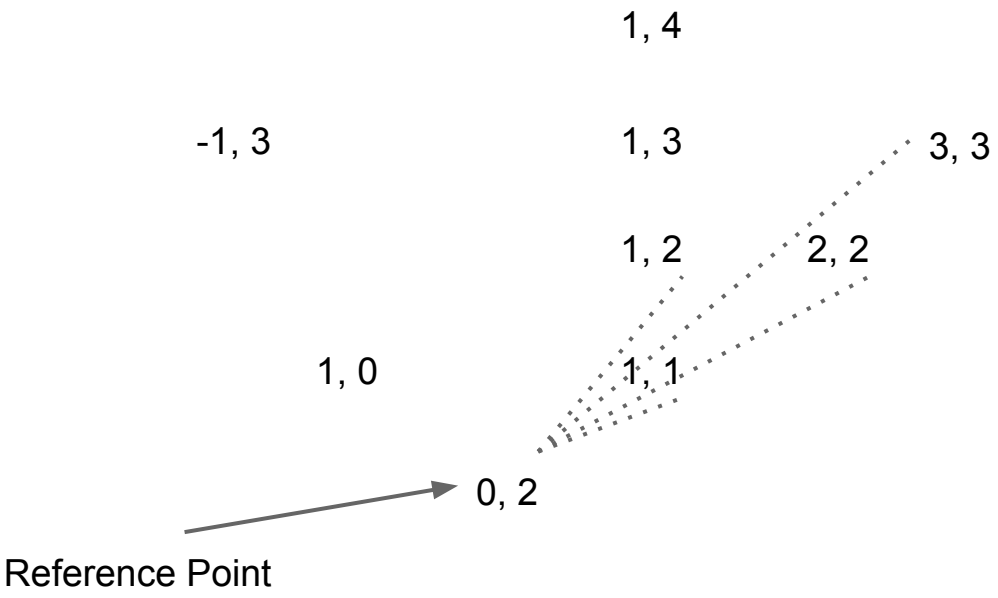
Convex Hull



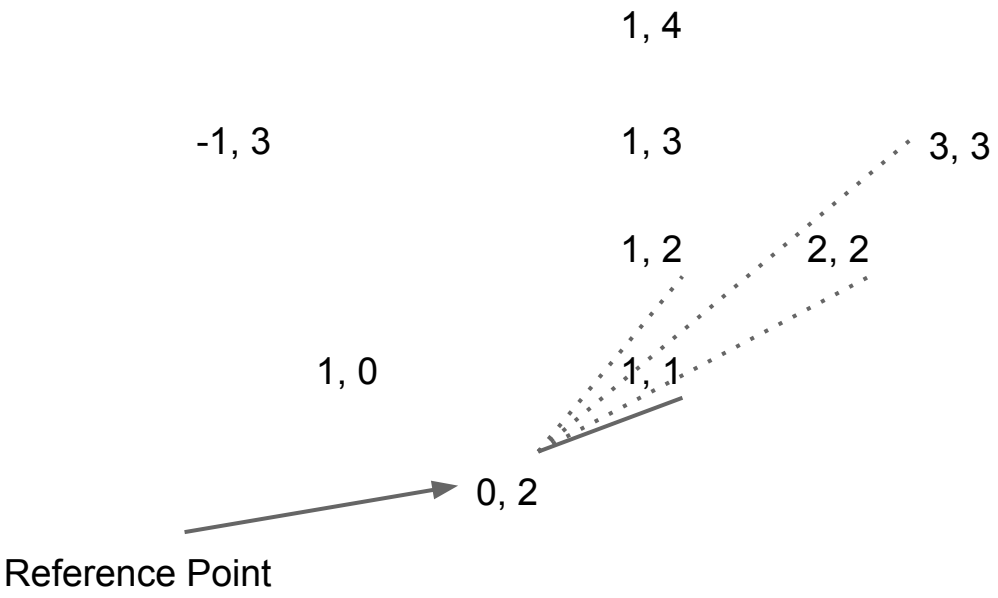
Convex Hull



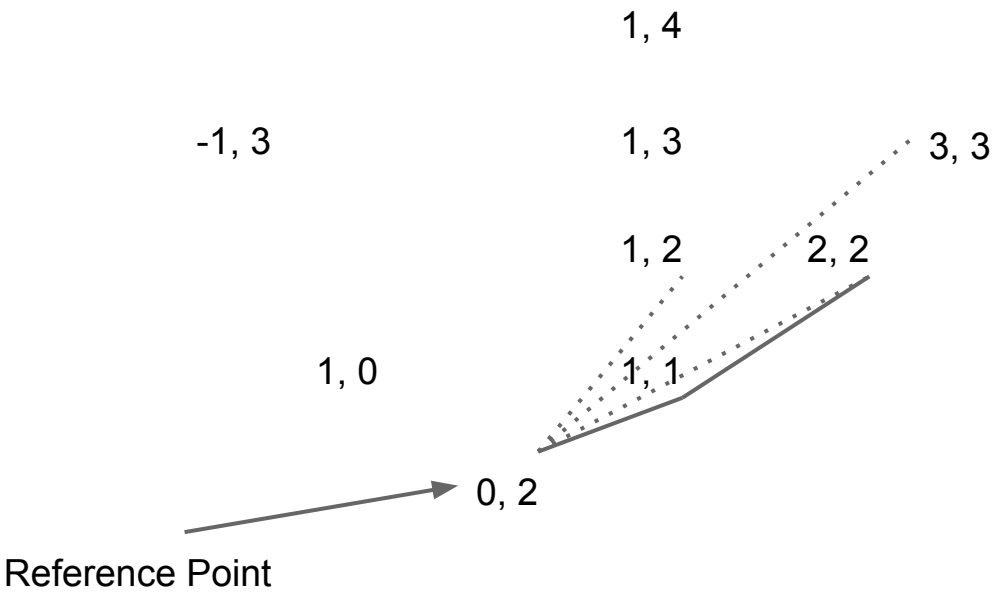
Convex Hull



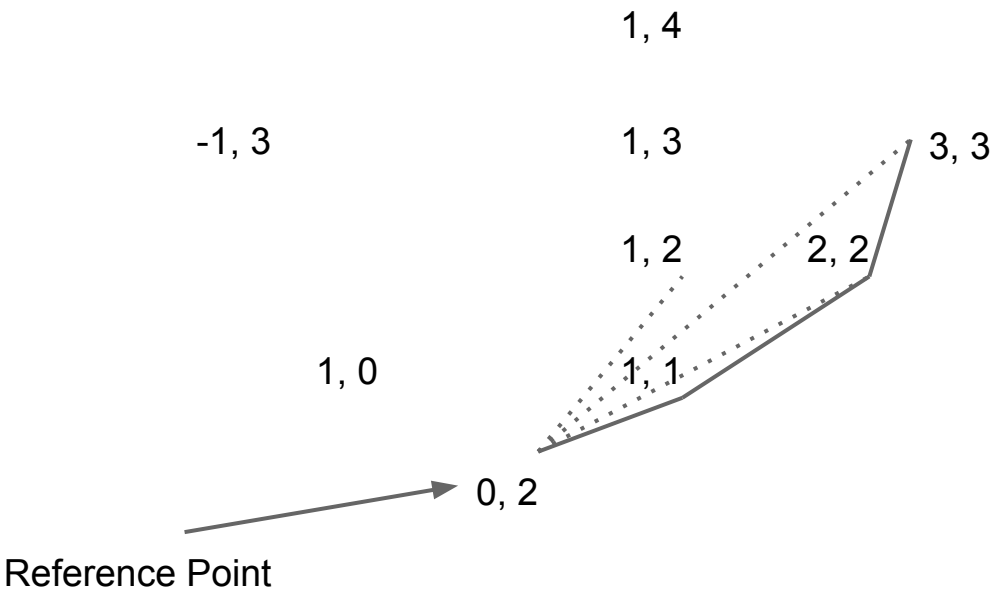
Convex Hull



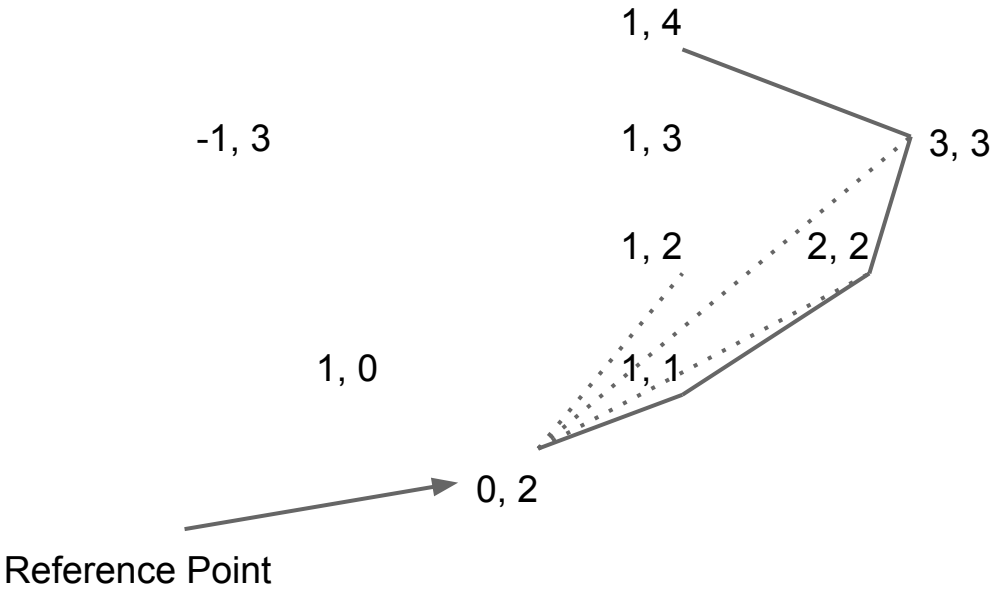
Convex Hull



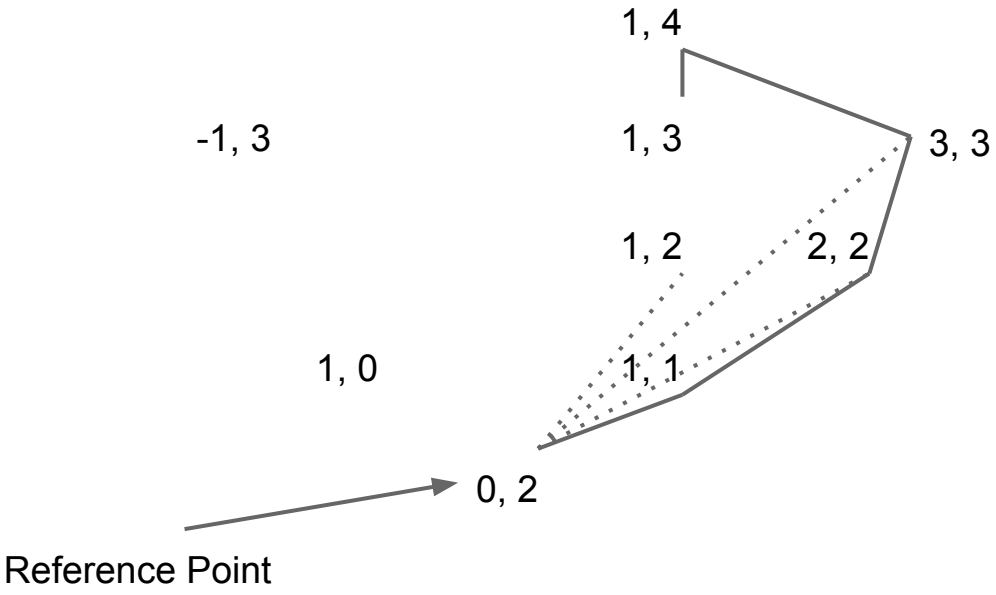
Convex Hull



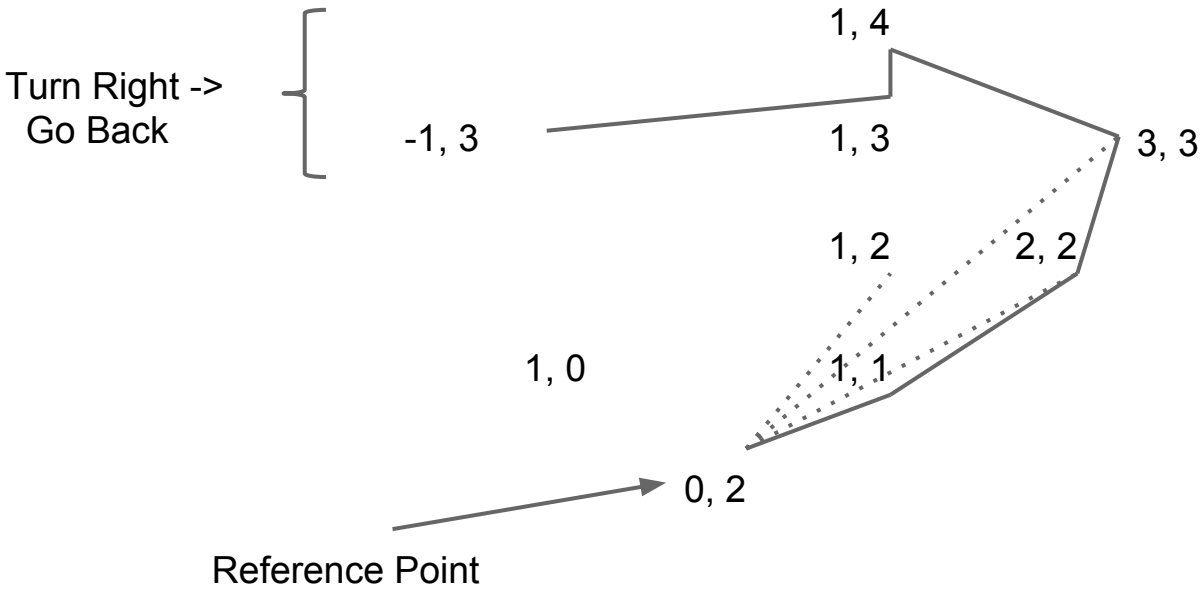
Convex Hull



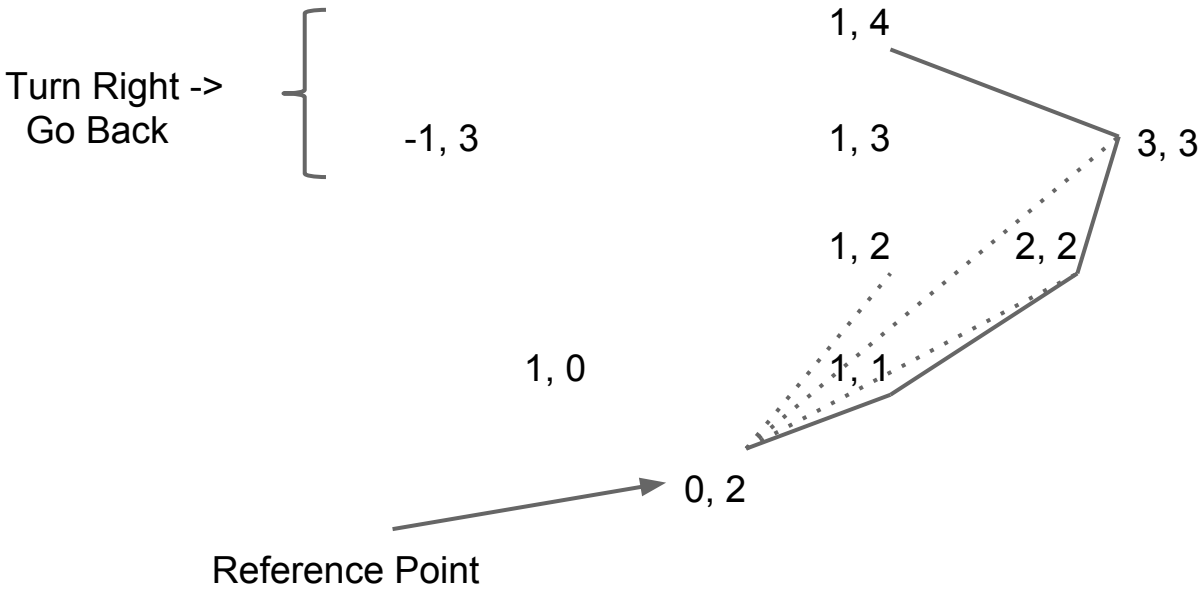
Convex Hull



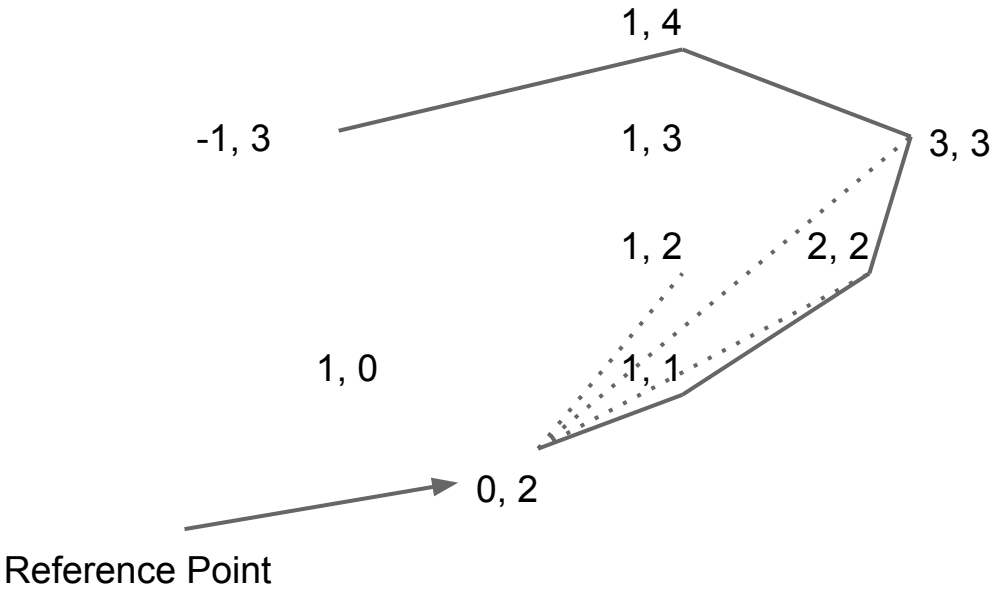
Convex Hull



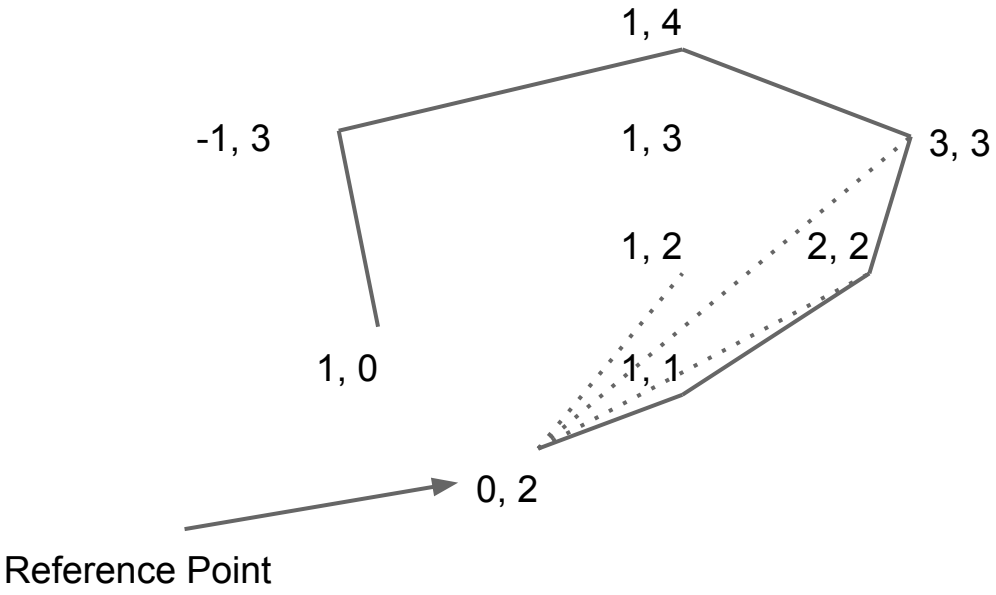
Convex Hull



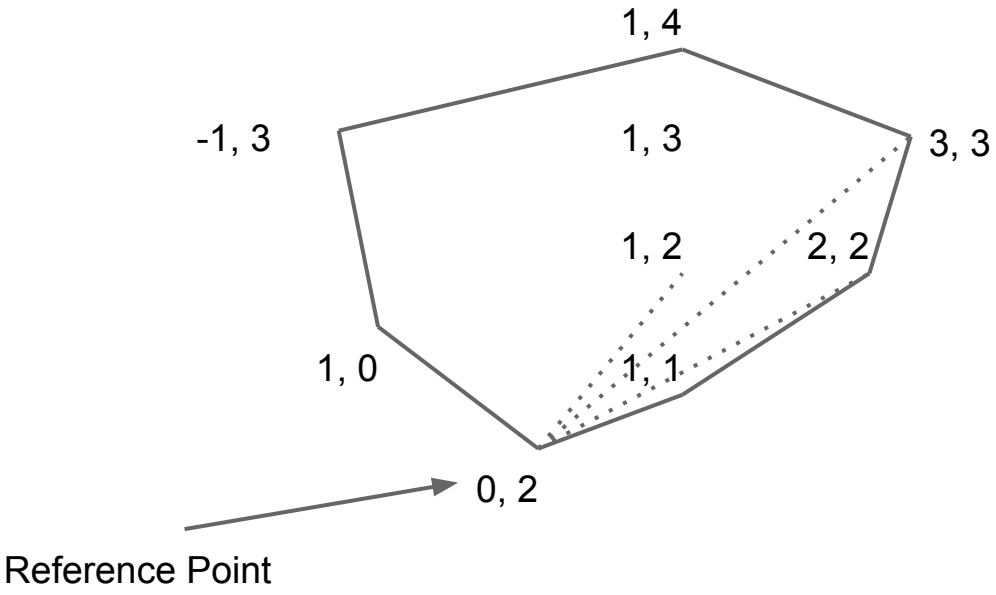
Convex Hull



Convex Hull



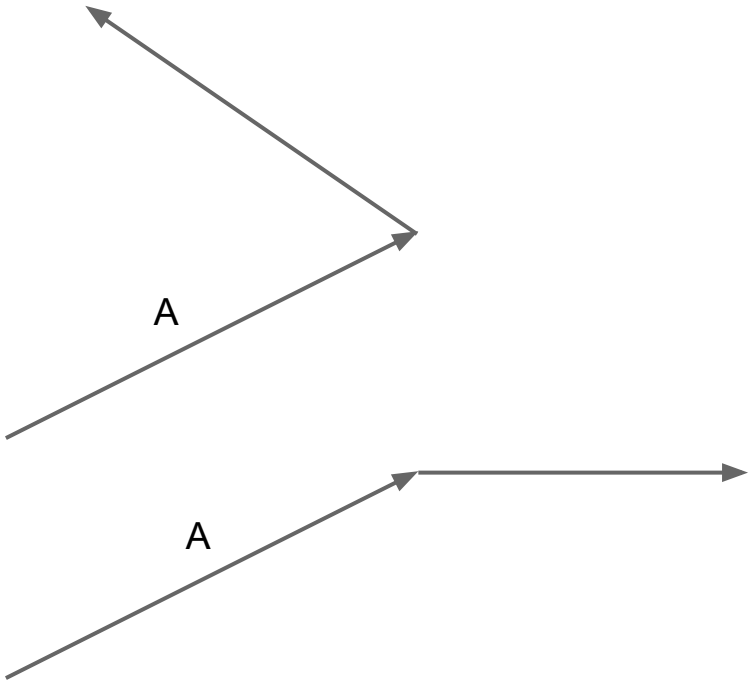
Convex Hull



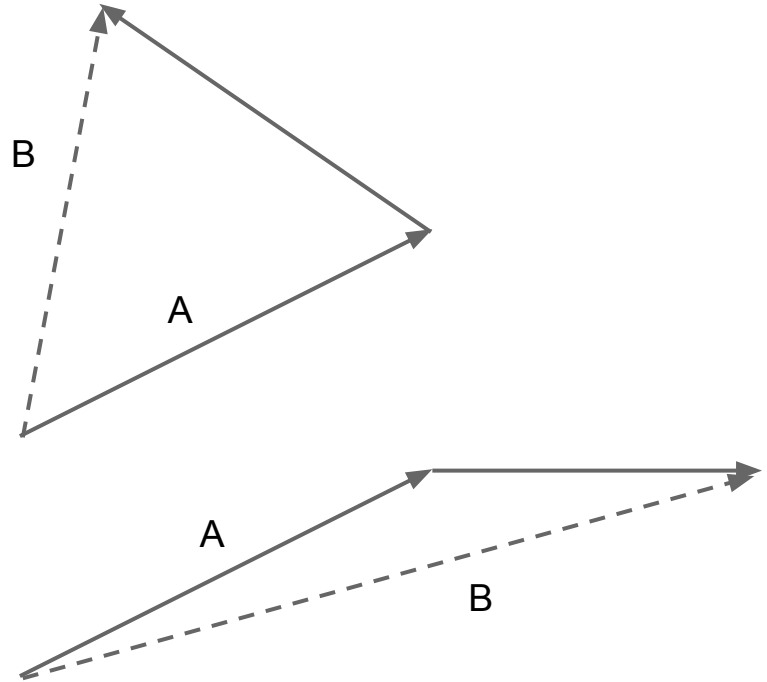
Graham Scan

1. Start with lowest point
2. Sort all points according to the angle they make with this point
3. Traverse points in angle order
   a. keep a list of current points in the hull
   b. add point to list if “left turn”
   c. remove points until a “left turn” is possible

Graham Scan: Find Direction



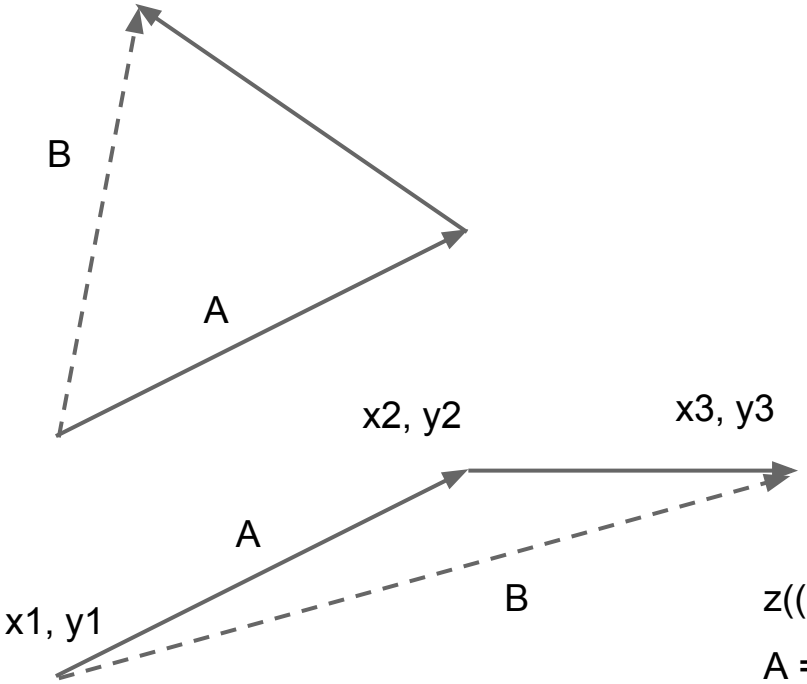
Graham Scan: Find Direction



Use cross product:

1. $z(A \times B) > 0 \Rightarrow$ going left
2. $z(A \times B) < 0 \Rightarrow$ going right

Graham Scan: Find Direction



Use cross product:

1. $z(A \times B) > 0 \Rightarrow$ going left
2. $z(A \times B) < 0 \Rightarrow$ going right

$z((u1, u2, u3) \times (v1, v2, v3)) = u1v2 - u2v1$

$A = (x2 - x1, y2 - y1, 0), B = (x3 - x1, y3 - y1, 0)$

$\Rightarrow$ **$z(A \times B) = (x2 - x1)(y3 - y1) - (x3 - x1)(y2 - y1)$**

Questions?