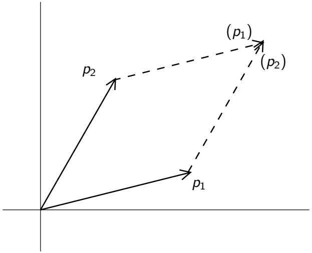
Game plan for computational geometry (CLRS 33)

- Basic computational geometry concepts (§33.1)
  - Representing points and segments
  - Testing for direction between two directed segments
  - Testing for intersection
- Computing the convex hull (§33.3)

Cross products, $p_1 = (x_1, y_1), p_2 = (x_2, y_2)$.

1. Actual cross product, $p_1 = (x_1, y_1, 0), p_2 = (x_2, y_2, 0)$. And so $p_1 \times p_2 = (0, 0, x_1y_2 - y_1x_2)$

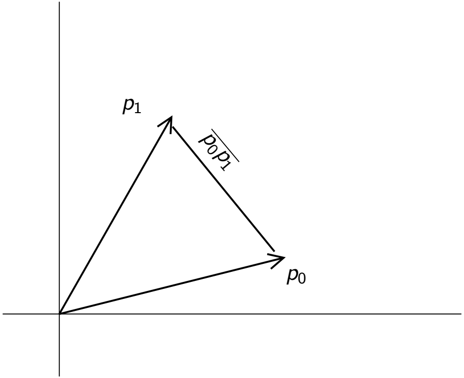
2. Area of a parallelogram

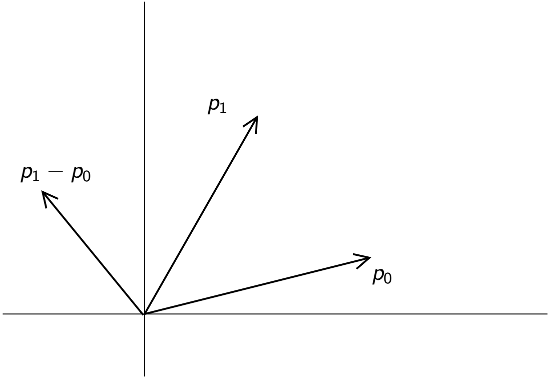


3. Determinant of a matrix

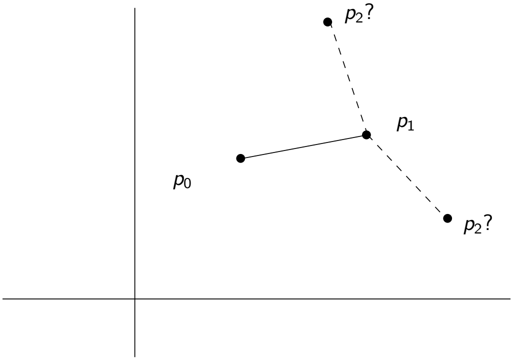
$$\det \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} = x_1y_2 - x_2y_1$$

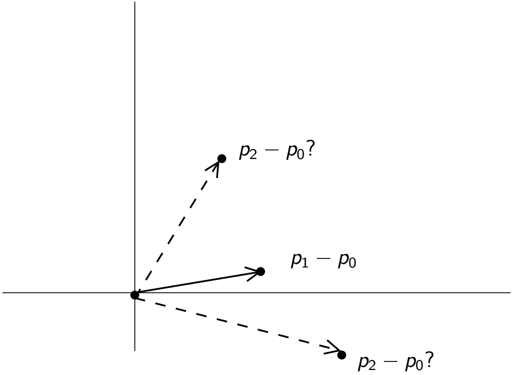
Subtraction

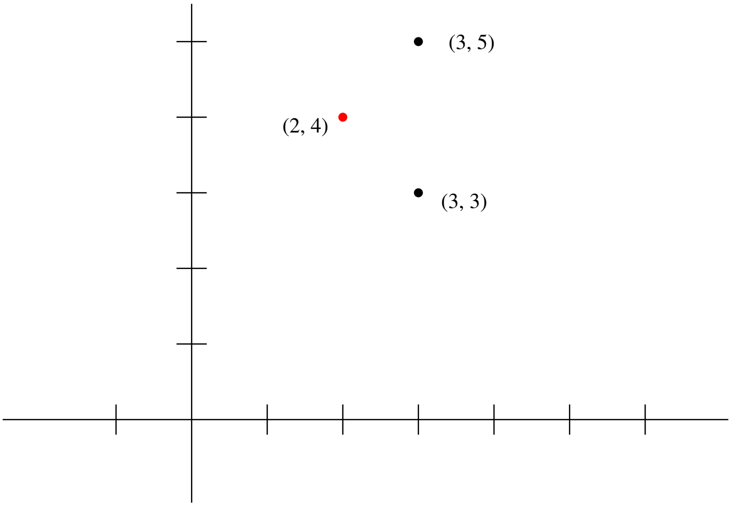




Subtraction to determine direction

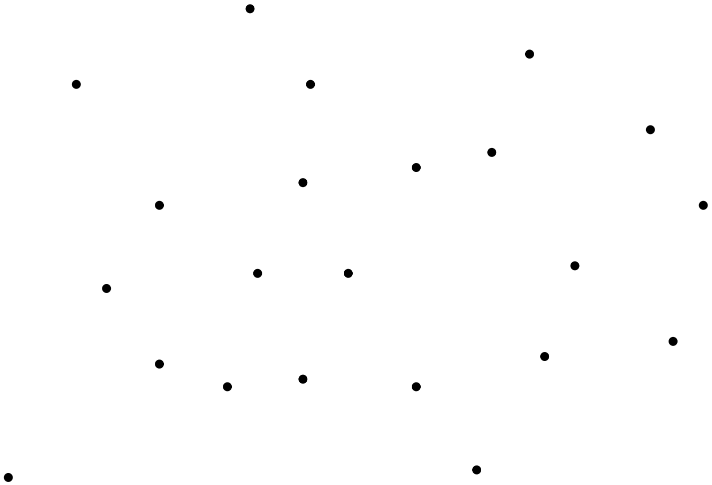





(3, 5)
(2, 4)
(3, 3)

Graham's Scan

- Find an extreme point, such as with a minimum y-value, call it p_0.



Theorem 1. The points with minimum and maximum x and y values are on the hull.

Proof. *WOLOG, consider the point (x_0, y_0) with minimum y value, and suppose that point is in the interior of the hull.*

Consider the vertical line through that point, and consider the segment of the convex hull that is intersected by that line below the point.

Call the point of intersection (x_1, y_1). Note that $x_1 = x_0$ and $y_1 < y_0$.

At least one end point of that segment must have a y value less than or equal to y_1, call that point (x_2, y_2). Then

$$y_2 \leq y_1 < y_0$$

...which contradicts (x_0, y_0) being the point with minimum y-value. $\square$

Graham's Scan

- Find an extreme point, such as with a minimum y-value, call it p_0.
- Sort the points by polar angle about p_0 (say, counterclockwise), call them $p_1, p_2, \ldots p_m$.
- Push p_0, p_1, and p_2 onto a stack.
- For each remaining point p_x in sorted order,
  - While the top stack elements make a non-left turn to p_i
    - pop
  - push p_x

Theorem 2. $\overline{p_0 p_1}$ makes a left turn to all other points.

Proof. *Suppose* $p_x \in \{p_2, p_3, \ldots p_m\}$*. Since the points are sorted by polar angle about* p_0*,* $(p_x - p_0) \times (p_1 - p_0) > 0$*, that is,* $\angle p_0 p_1 p_x$ *turns left.* $\square$

Corollary 3. The algorithm never pops p_1.

Proof. *By Theorem 2, the guard of the while loop is never true when* p_1 *is on the top of the stack.* $\square$

Theorem 4. The points with least and greatest polar angles with respect to p_0 are on the convex hull.

Proof. *Note that* p_1 *is the point with least polar angle with respect to* p_0*, since the points are sorted by polar angle. Suppose, then, that* p_1 *is in the interior of the hull.*

Similarly to the proof of Theorem 1, we shall consider the segment of the hull intersected by a line through p_1*. The point* p_0 *is either left or right of* p_1*.*

Case 1. *Suppose* p_0 *is left of* p_1*. Then consider the* **vertical** *line through* p_1 *that hits a segment (or point) below* p_1*. At least one endpoint of that segment (or the point itself) must have a smaller polar angle from* p_0 *than* p_1 *has, which contradicts the sortedness of the points.*

Case 2. *Suppose* p_0 *is right of* p_1*. Then consider the* **horizontal** *line through* p_1 *that hits a segment (or point) right of* p_1*. At least one endpoint of that segment (or the point itself) must have a smaller polar angle from* p_0 *than* p_0 *has, which contradicts the sortedness of the points.*

The reasoning is similar for the point that has the greatest polar angle, p_m*.* □

Postcondition (correctness claim). The contents of the stack are the points of the hull, counterclockwise.

To prove this, we using the following as a lemma:

Invariant. After i iterations of the for loop, the stack contains the convex hull of the points $\{p_0, p_1, \dots p_{i+2}\}$.

> **Initialization.** *Before the loop begins, the stack contains p_0, p_1, p_2, which constitute the convex hull of those points since any three non-collinear points are the end points of their own convex hull.*
>
> **Maintenance.** *Suppose after i iterations, the stack contains the convex hull of $\{p_0, p_1, \dots, p_{i+2}\}$. Consider what happens on the next iteration, on which we process the point p_{i+3}.*
>
> *Let p_j be the top point on the stack at the end of the while loop. The stack is in the same state as after the $j-2$nd iteration.*

... wait, how do we know that?

Invariant. After i iterations of the for loop, the stack contains the convex hull of the points $\{p_0, p_1, \ldots p_{i+2}\}$.

> **Initialization.** *Before the loop begins, the stack contains p_0, p_1, p_2, which constitute the convex hull of those points since any three non-collinear points are the end points of their own convex hull.*
>
> **Maintenance.** *Suppose after i iterations, the stack contains the convex hull of $\{p_0, p_1, \ldots, p_{i+2}\}$. Consider what happens on the next iteration, on which we process the point p_{i+3}.*
>
> *Let p_j be the top point on the stack at the end of the while loop. The stack is in the same state as after the $j-2$nd iteration. By Corollary 3, the stack in that state had size at least 2. Let p_k be the point immediately below p_j.*
>
> *By induction, the stack in that state contains the convex hull of $\{p_0, p_1, \ldots p_k, p_j\}$*

Something's wrong ...

Invariant. After i iterations of the for loop, the stack contains the convex hull of the points $\{p_0, p_1, \ldots p_{i+2}\}$.

Initialization. *Before the loop begins, the stack contains* p_0, p_1, p_2*, which constitute the convex hull of those points since any three non-collinear points are the end points of their own convex hull.*

Maintenance. *Suppose that for some* $i \geq 0$*, for all* j*,* $0 \leq j \leq i$*, after* j *iterations, the stack contains the convex hull of* $\{p_0, p_1, \ldots, p_{j+2}\}$*.*

Consider what happens on the iteration after i *iterations, on which we process the point* p_{i+3}*.*

Let p_j *be the top point on the stack at the end of the while loop. The stack is in the same state as after the* $j-2$*nd iteration. By Corollary 3, the stack in that state had size at least 2. Let* p_k *be the point immediately below* p_j*.*

By **strong** *induction, the stack in that state contains the convex hull of* $\{p_0, p_1, \ldots p_k, p_j\}$*.*

[Need to show that the points p_{j+1} to p_{i_2}, inclusive, are on the interior of the hull of $\{p_0, p_1, \ldots p_{i+3}$.]

Suppose $p_t \in \{p_{j+1}, p_{i_2}\}$. Consider various cases corresponding to when p_t was popped from the stack.

Case 1. *Suppose p_t was popped on the current iteration. What was immediately below p_t on the stack?*

Case 1a. *Suppose p_j was immediately below p_t on the stack . . .*

Case 1b. *Suppose another (interior) point, p_r was below p_t on the stack . . .*

Case 2. *Suppose p_t was popped on another iteration of the for loop, when p_s was being considered. Let p_q be the point immediately below p_t on the stack at that time . . .*

□

Graham's Scan

- Find an extreme point, such as with a minimum y-value, call it p_0.
- Sort the points by polar angle about p_0 (say, counterclockwise), call them $p_1, p_2, \ldots p_m$.
- Push p_0, p_1, and p_2 onto a stack.
- For each remaining point p_x in sorted order,
  - While the top stack elements make a non-left turn to p_i
    - pop
  - push p_x

Termination. *After $m - 2$ iterations, the for loop exists, and, by the loop invariant, the stack is the convex hull of the points $\{p_0, p_1, \ldots p_m\}$.* □