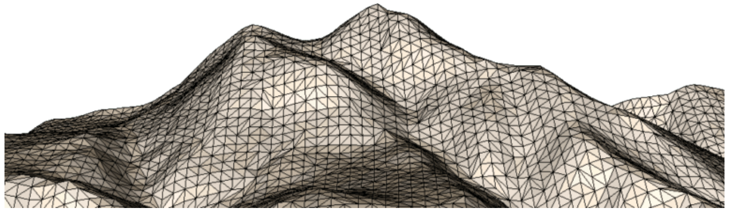
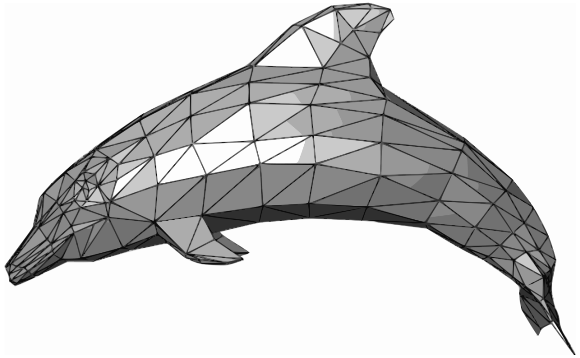
ICS 621: Analysis of Algorithms

Prof. Nodari Sitchinava



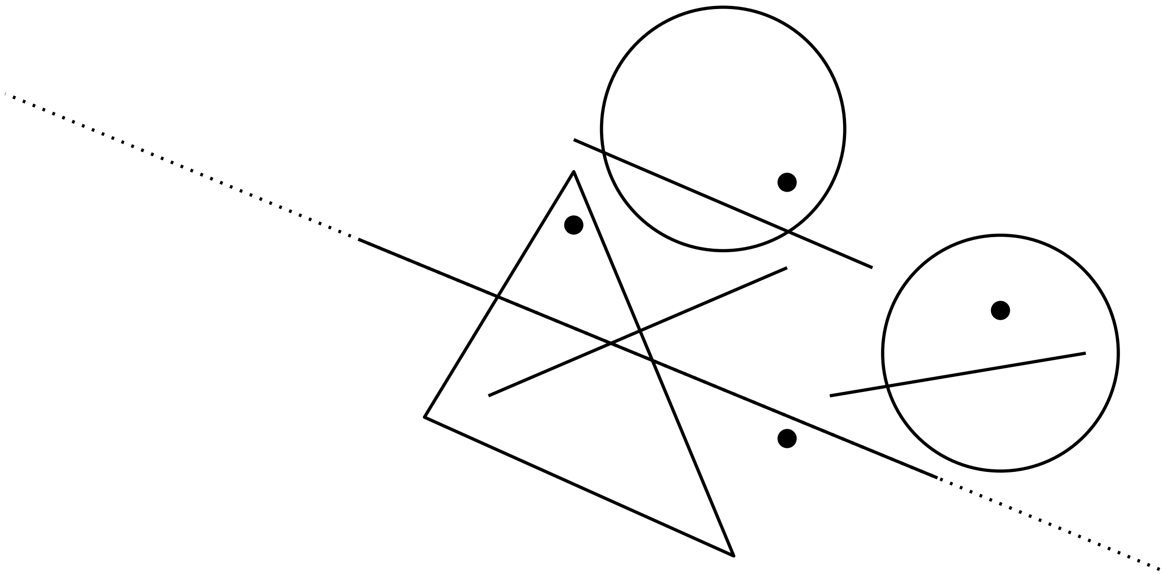


Computation Geometry

Computation Geometry

Computational problems on discrete geometric objects:

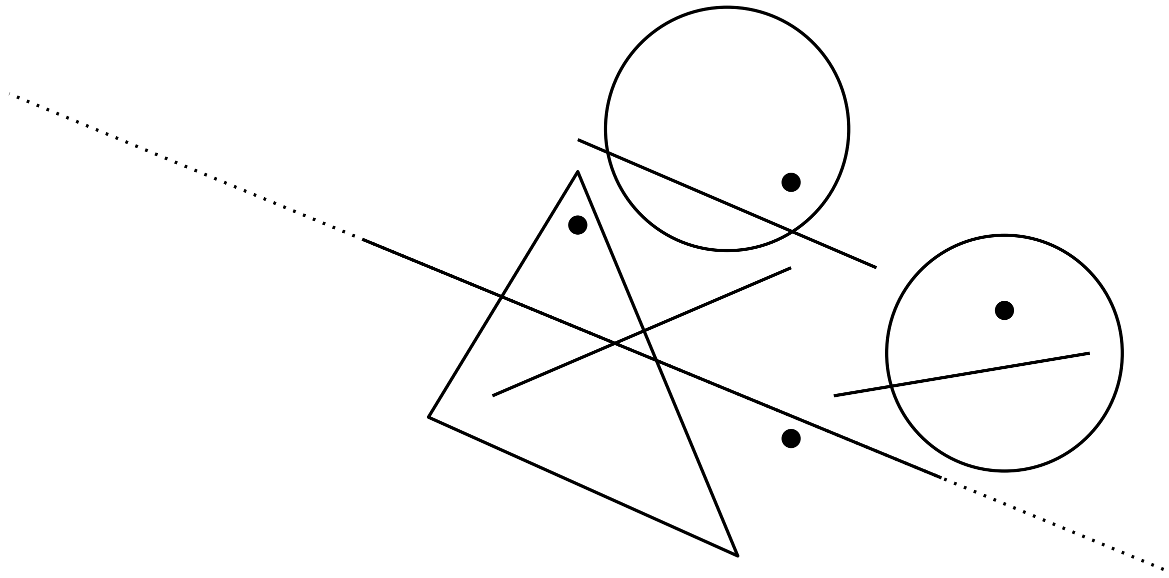
- Points
- Lines & line segments
- Planes
- Surfaces
- Circles/spheres



Computation Geometry

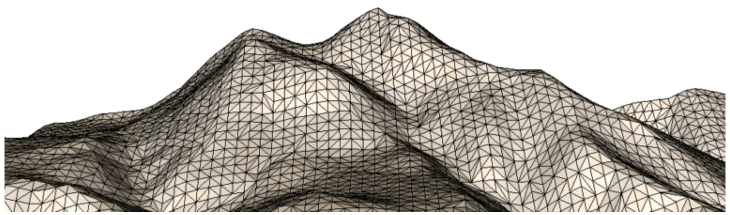
Computational problems on discrete geometric objects:

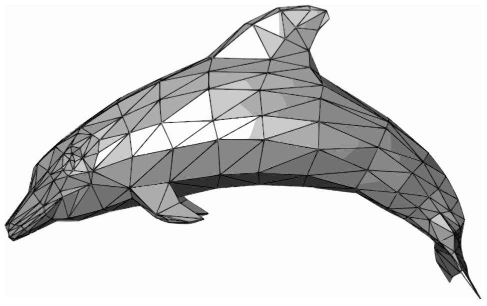
- Points
- Lines & line segments
- Planes
- Surfaces
- Circles/spheres



Applications:

- Computer graphics
- Geographic Information Systems (GIS)
- Computer Aided Design (CAD)
- Robotics
- ...

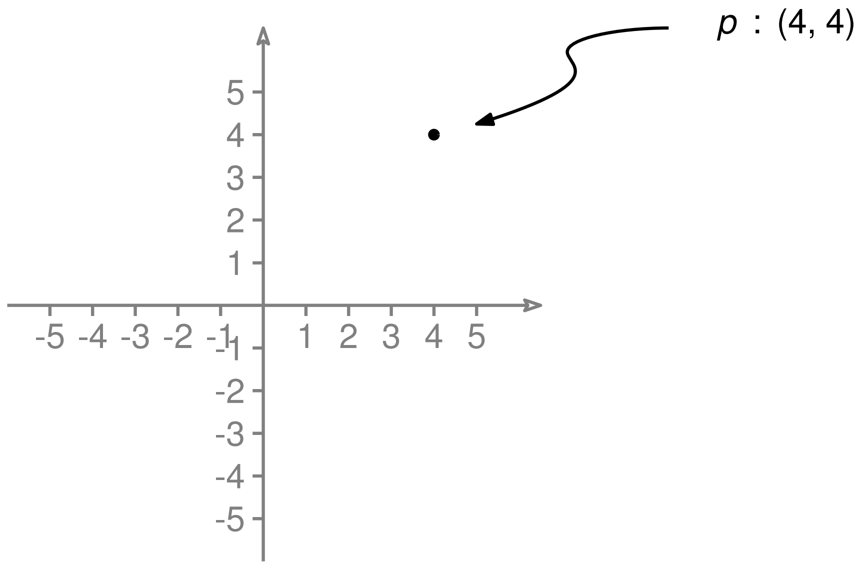




Data Representation (2-D)

- Points

  $p : (a, b)$



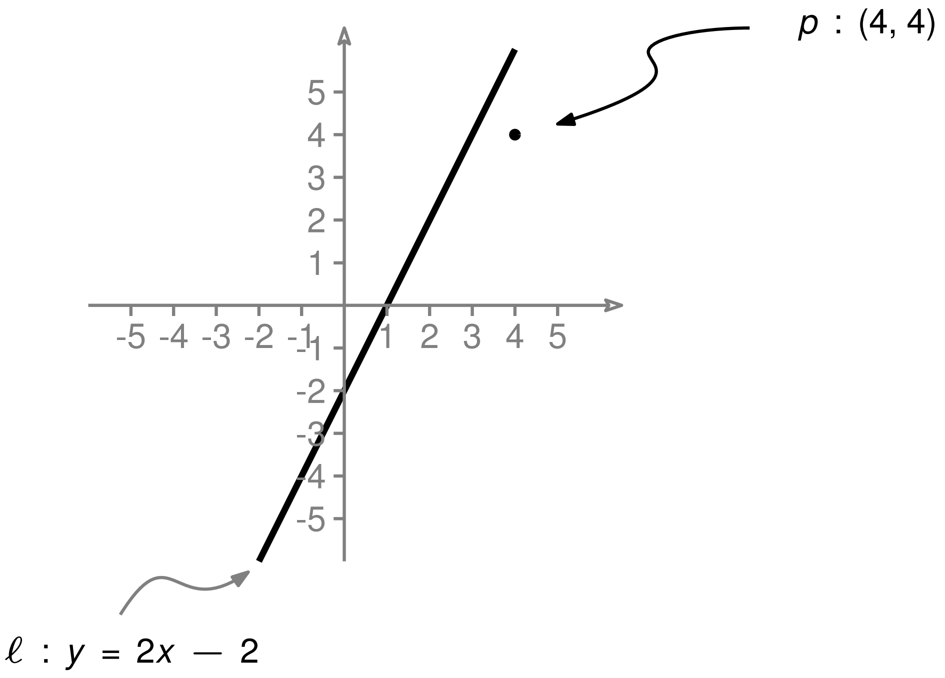
Data Representation (2-D)

- Points

  $p : (a, b)$

- (Non-vertical) lines

  $\ell : y = mx + b$



Data Representation (2-D)

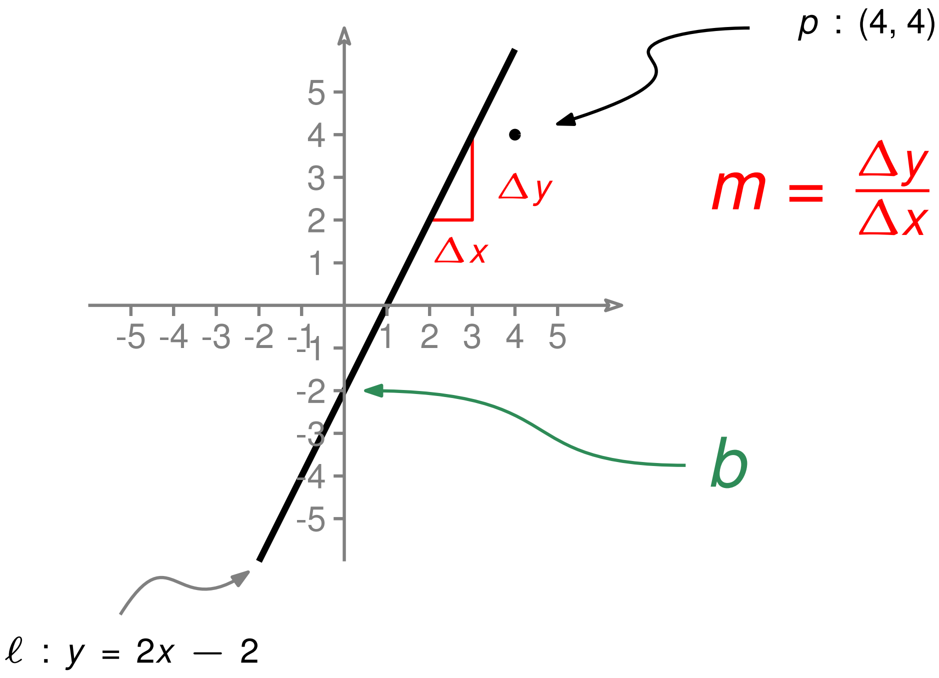
- Points

  $p : (a, b)$

- (Non-vertical) lines

  $\ell : y = mx + b$

  slope *y*-intercept



Data Representation (2-D)

Representation

- Points

  $p : (a, b)$

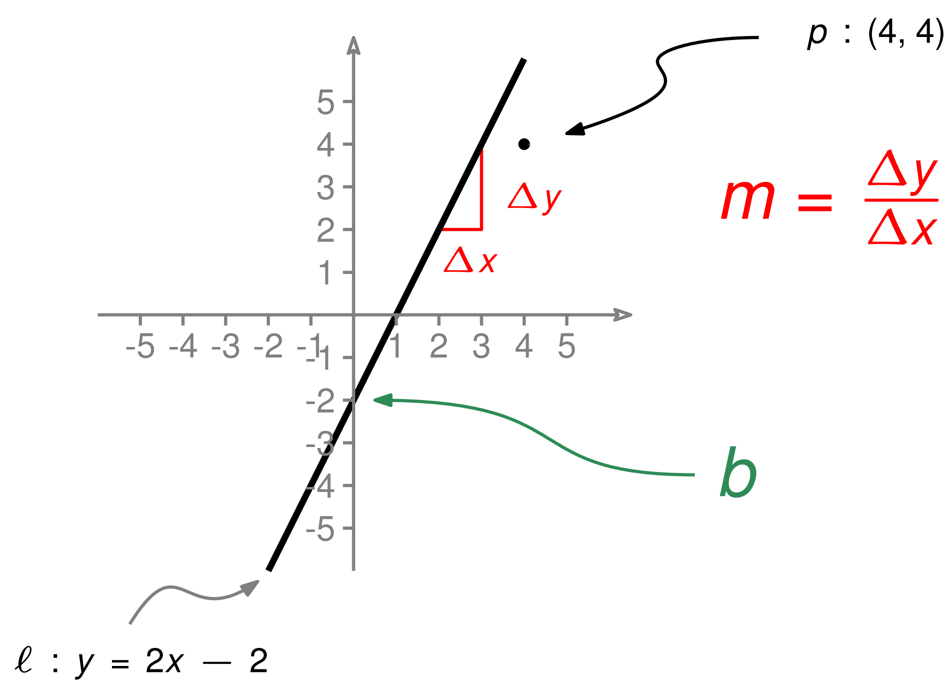
  $$\begin{pmatrix} a \\ b \end{pmatrix}$$

- (Non-vertical) lines

  $$\begin{pmatrix} m \\ b \end{pmatrix}$$

  $\ell : y = mx + b$

  slope (m) — y-intercept (b)



Data Representation (2-D)

Representation

- Points

  $p : (a, b)$

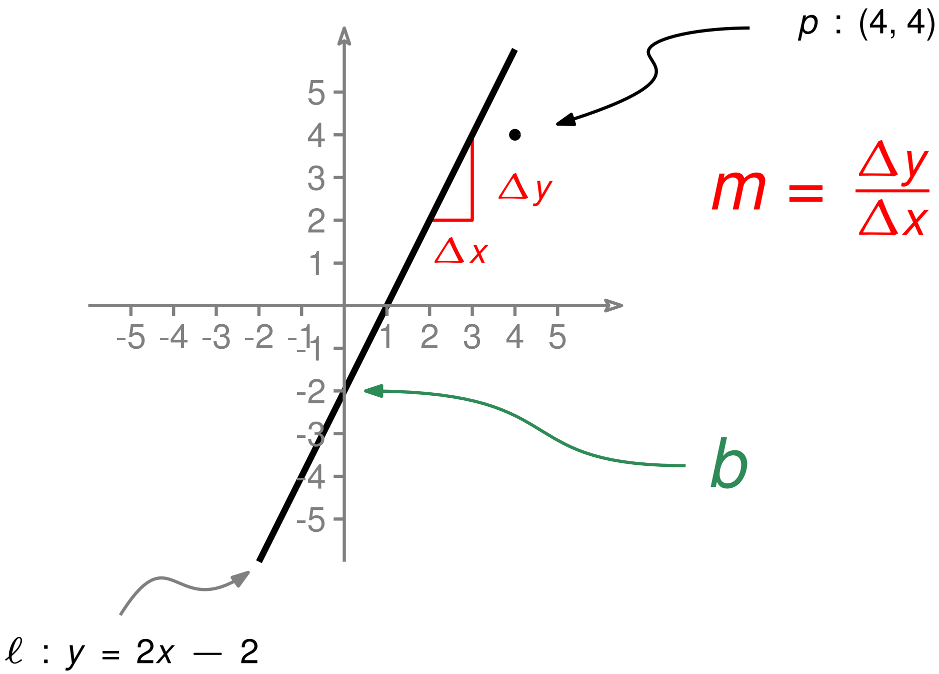
  $$\begin{pmatrix} a \\ b \end{pmatrix}$$

- (Non-vertical) lines

  $$\begin{pmatrix} m \\ b \end{pmatrix}$$

  $\ell : y = mx + b$

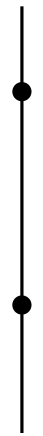
  slope y-intercept



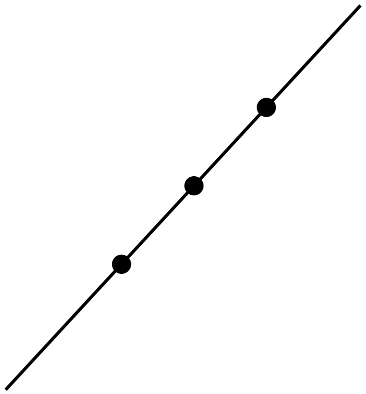
Point $p = (a, b)$ Line $\ell = (m, b)$

General Position Assumption

- No two points have the same *y*-coordinates (no vertical lines)

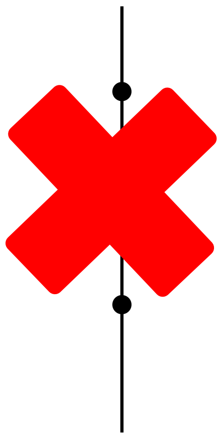


- No three points are co-linear

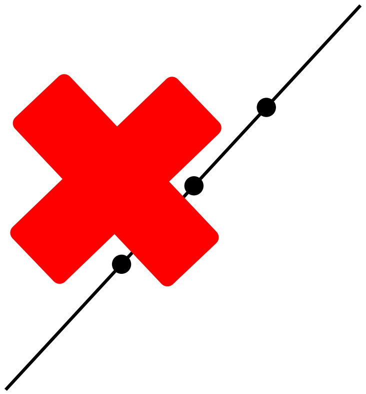


General Position Assumption

- No two points have the same *y*-coordinates (no vertical lines)

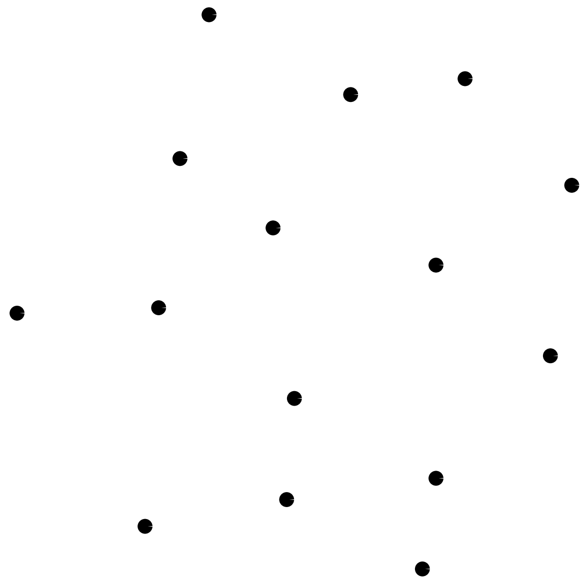


- No three points are co-linear



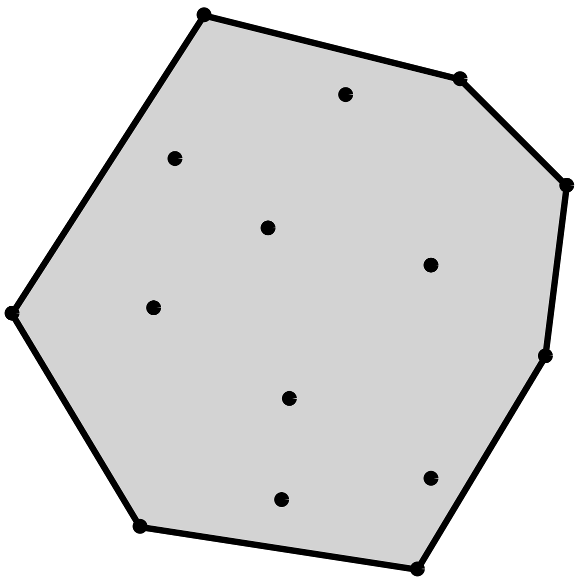
Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



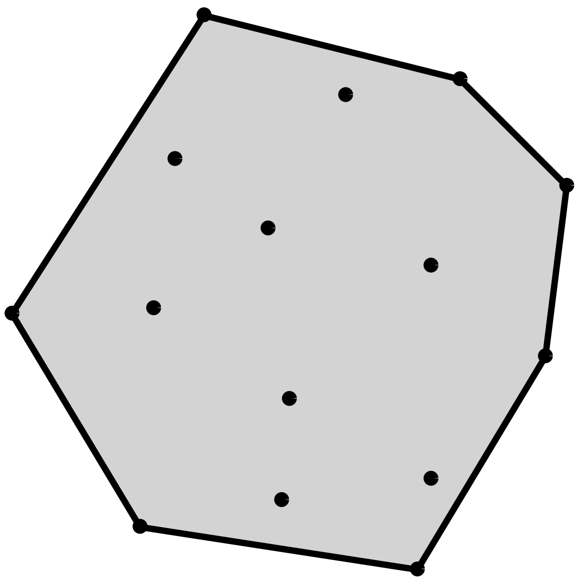
Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Convex Hull

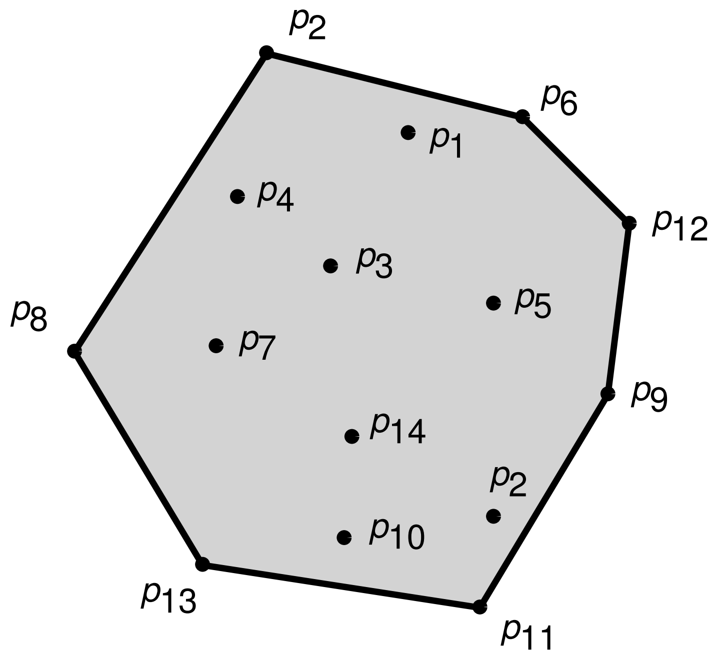
- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Polygon: circular sequence of points connected by line segments

Convex Hull

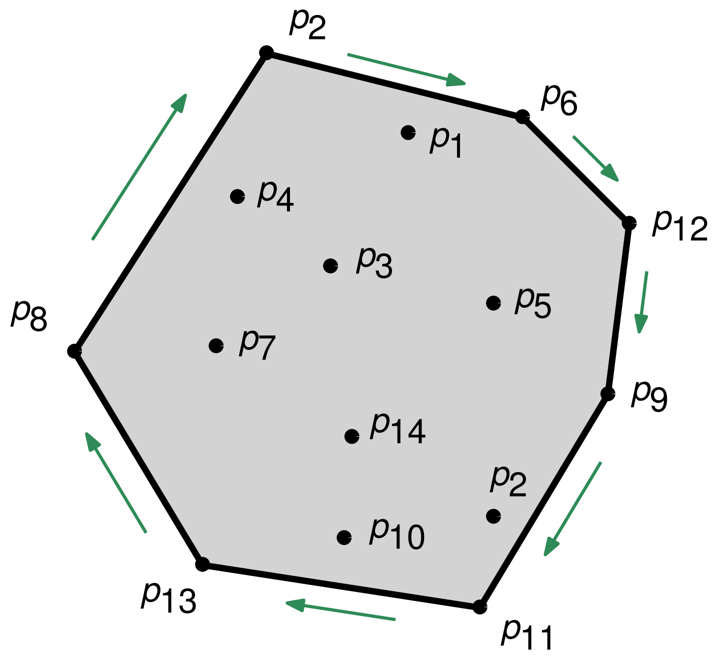
- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Polygon: circular sequence of points connected by line segments

Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



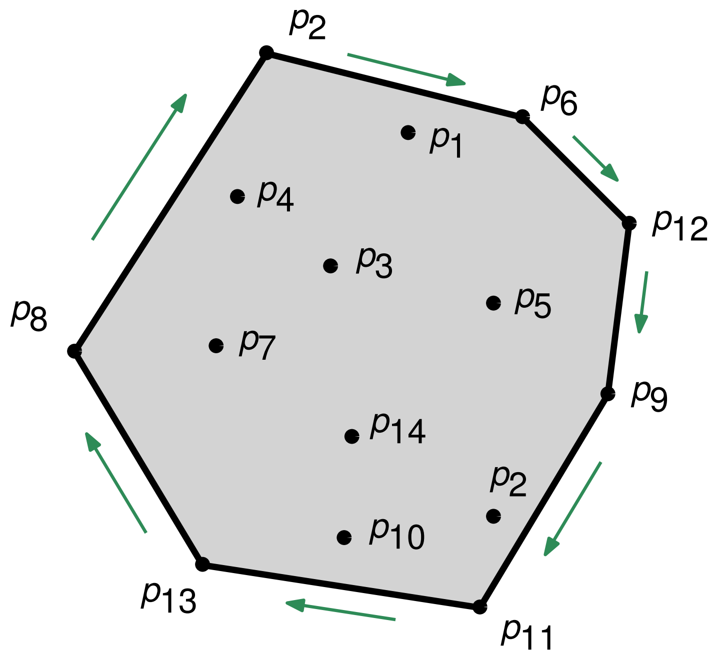
Output:

$p_2, p_6, p_{12}, p_9, p_{11}, p_{13}, p_8$

Polygon: circular sequence of points connected by line segments

Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Output:

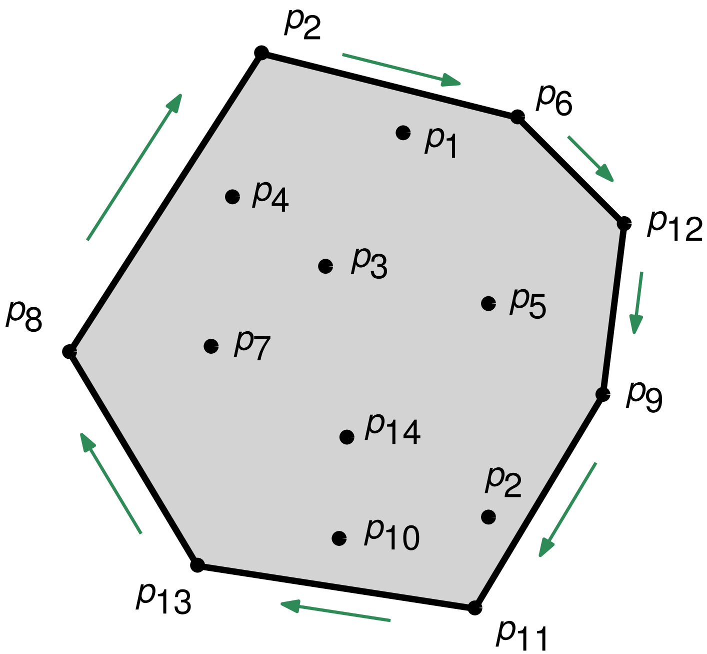
$p_2, p_6, p_{12}, p_9, p_{11}, p_{13}, p_8$

Polygon: circular sequence of points connected by line segments

S is convex iff for every pair of points $p, q \in S$, the segment $\overline{pq}$ is in S.

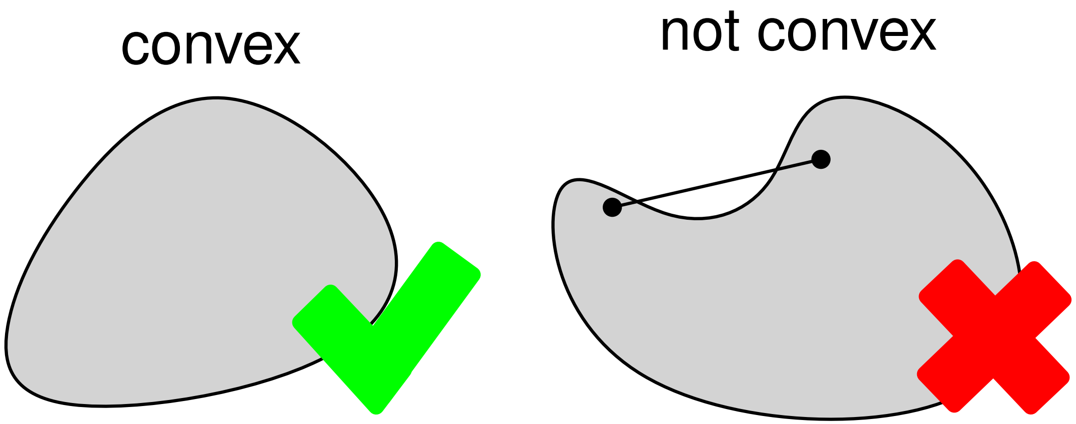
Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Output:

$p_2, p_6, p_{12}, p_9, p_{11}, p_{13}, p_8$

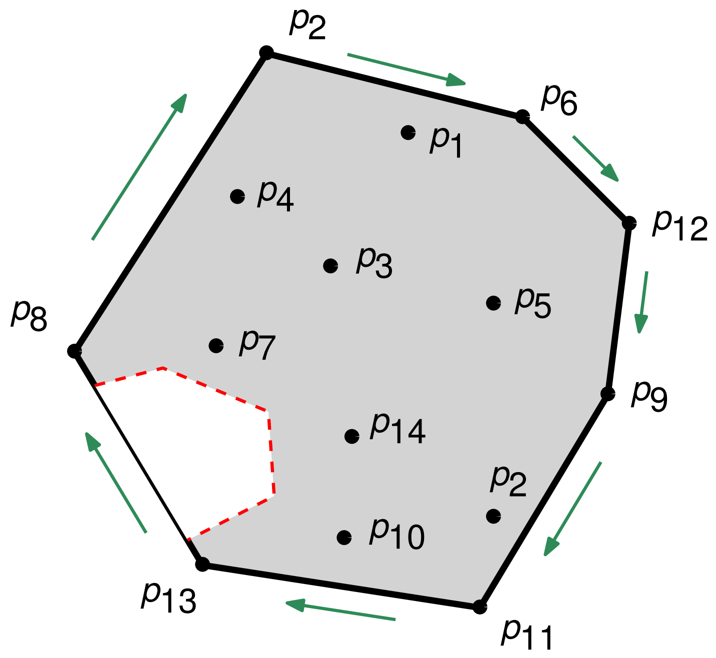


Polygon: circular sequence of points connected by line segments

S is convex iff for every pair of points $p, q \in S$, the segment $\overline{pq}$ is in S.

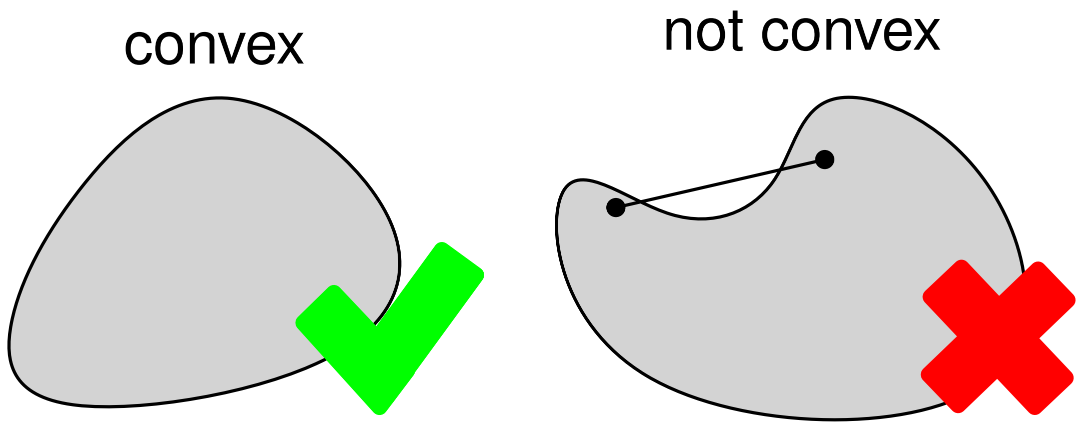
Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Output:

$p_2, p_6, p_{12}, p_9, p_{11}, p_{13}, p_8$

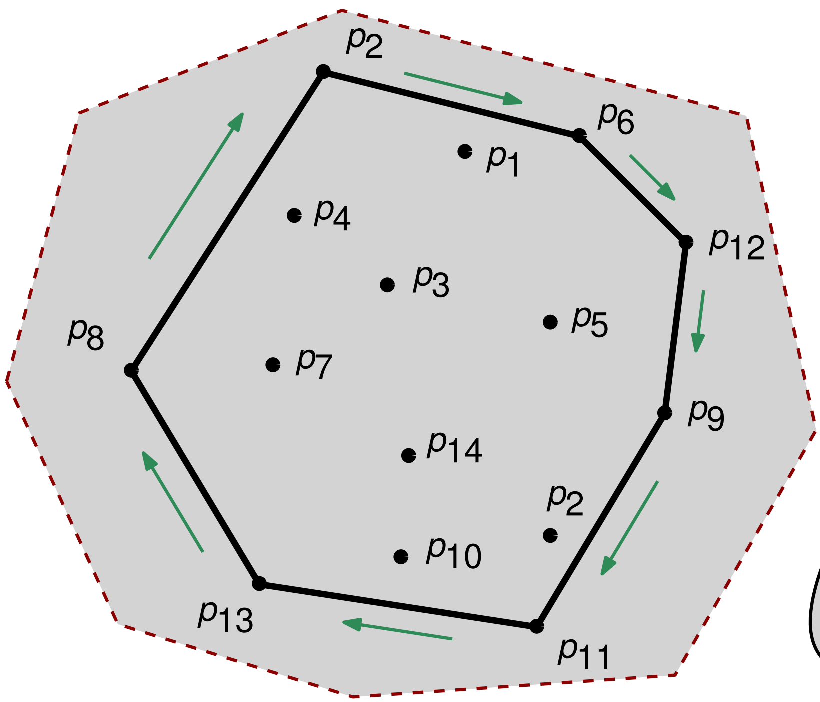


Polygon: circular sequence of points connected by line segments

S is convex iff for every pair of points $p, q \in S$, the segment $\overline{pq}$ is in S.

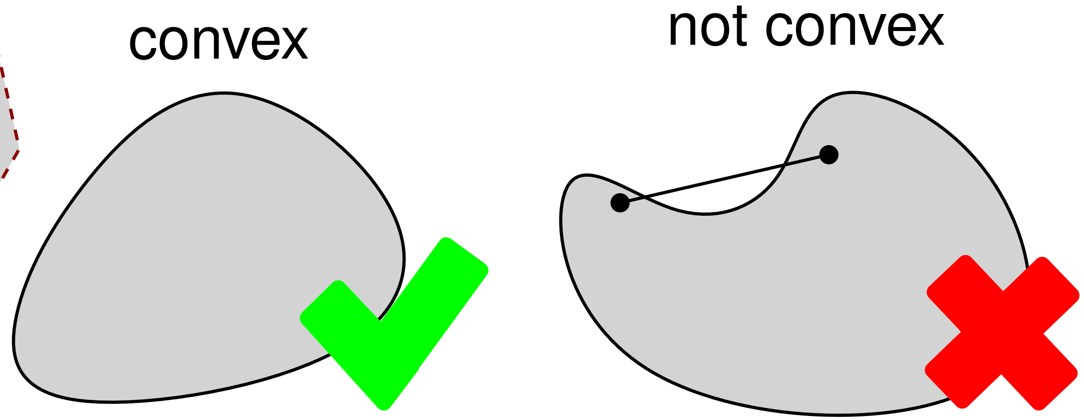
Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Output:

$p_2, p_6, p_{12}, p_9, p_{11}, p_{13}, p_8$



Polygon: circular sequence of points connected by line segments

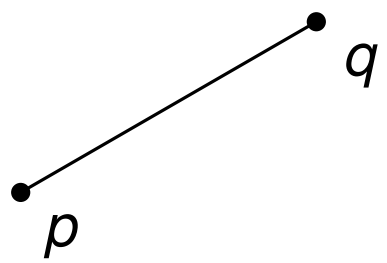
S is convex iff for every pair of points $p, q \in S$, the segment $\overline{pq}$ is in S.

Segment $\overline{pq}$

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$

Segment $\overline{pq}$

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$

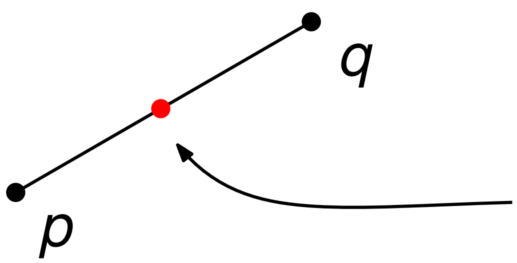


Definition. *The segment* $\overline{pq}$ *is the set of all points* $r = \alpha p + (1 - \alpha)q$ *for all* $0 \leq \alpha \leq 1$, *i.e.,*

$$r = \binom{r_x}{r_y} = \alpha \binom{p_x}{p_y} + (1 - \alpha) \binom{q_x}{q_y} = \binom{\alpha p_x + (1-\alpha) q_x}{\alpha p_y + (1-\alpha) q_y}$$

Segment $\overline{pq}$

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$



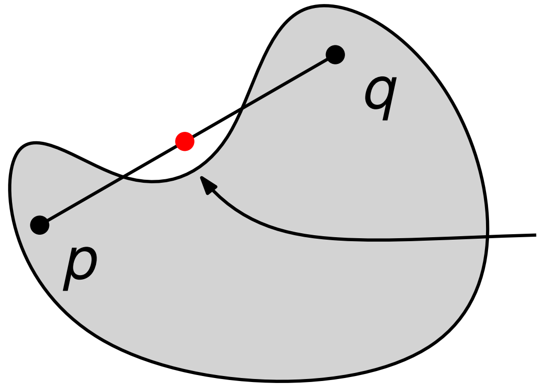
$$r = 0.5p + 0.5q = \binom{0.5p_x + 0.5q_x}{0.5p_y + 0.5q_y}$$

Definition. *The segment $\overline{pq}$ is the set of all points $r = \alpha p + (1 - \alpha)q$ for all $0 \leq \alpha \leq 1$, i.e.,*

$$r = \binom{r_x}{r_y} = \alpha\binom{p_x}{p_y} + (1 - \alpha)\binom{q_x}{q_y} = \binom{\alpha p_x + (1-\alpha)q_x}{\alpha p_y + (1-\alpha)q_y}$$

Segment $\overline{pq}$

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$



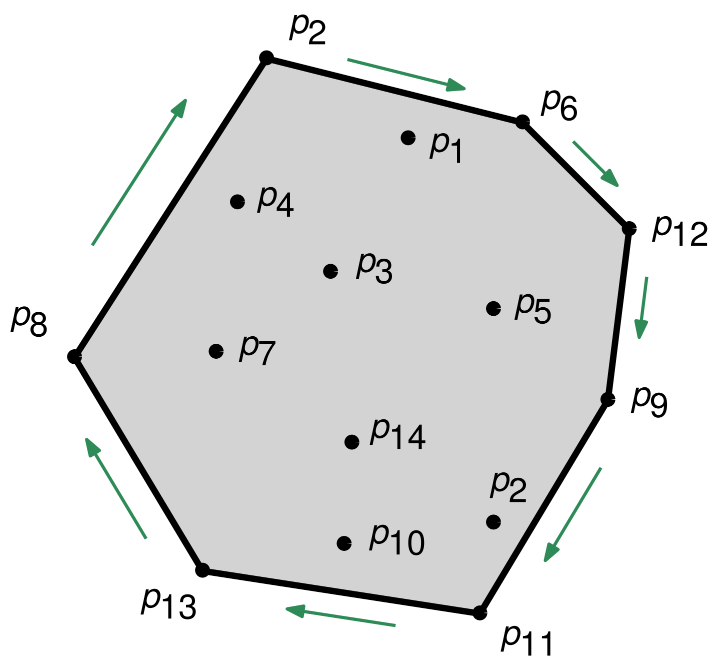
$r = 0.5p + 0.5q = \binom{0.5p_x + 0.5q_x}{0.5p_y + 0.5q_y}$

Definition. *The segment* $\overline{pq}$ *is the set of all points* $r = \alpha p + (1 - \alpha)q$ *for all* $0 \leq \alpha \leq 1$, *i.e.,*

$$r = \binom{r_x}{r_y} = \alpha \binom{p_x}{p_y} + (1 - \alpha) \binom{q_x}{q_y} = \binom{\alpha p_x + (1-\alpha) q_x}{\alpha p_y + (1-\alpha) q_y}$$

Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Output:

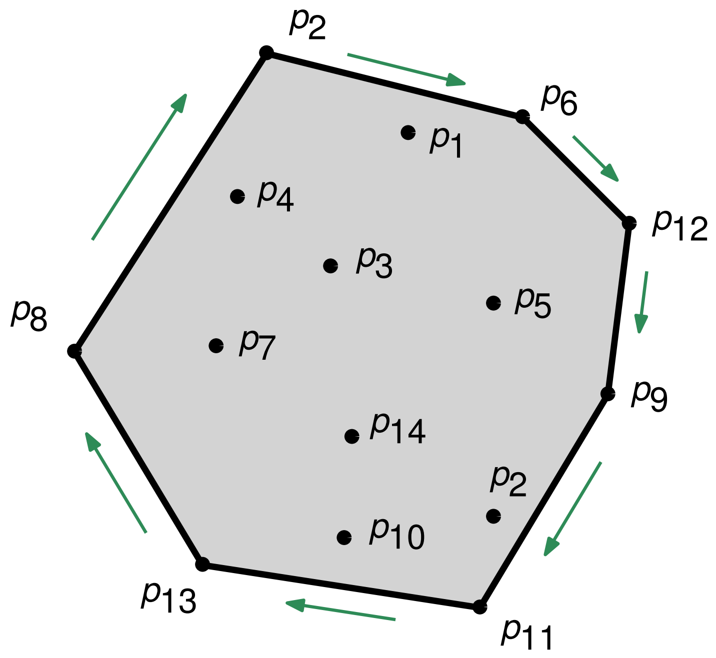
$p_2, p_6, p_{12}, p_9, p_{11}, p_{13}, p_8$

Polygon: circular sequence of points connected by line segments

S is convex iff for every pair of points $p, q \in S$, the segment $\overline{pq}$ is in S.

Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Output:

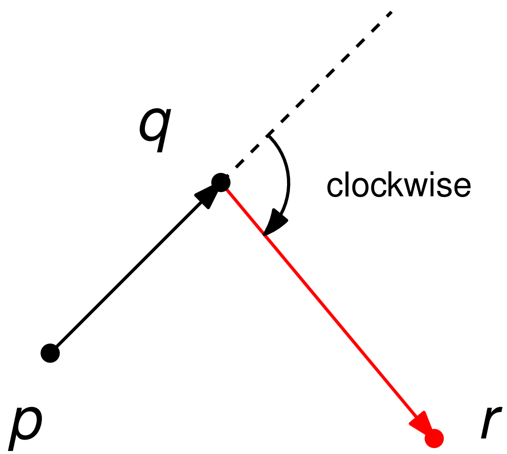
$p_2, p_6, p_{12}, p_9, p_{11}, p_{13}, p_8$

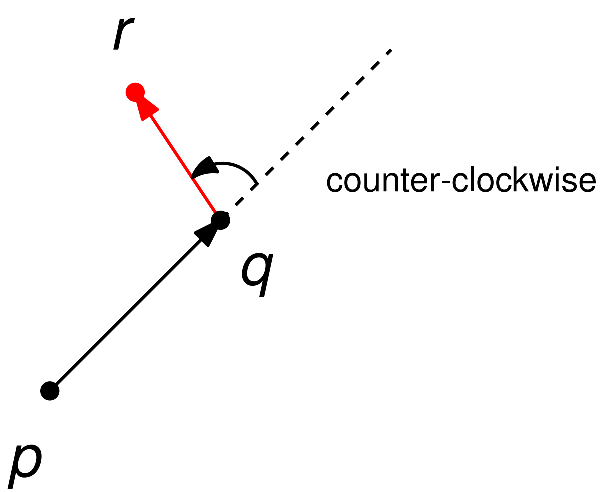
In clockwise order

Polygon: circular sequence of points connected by line segments

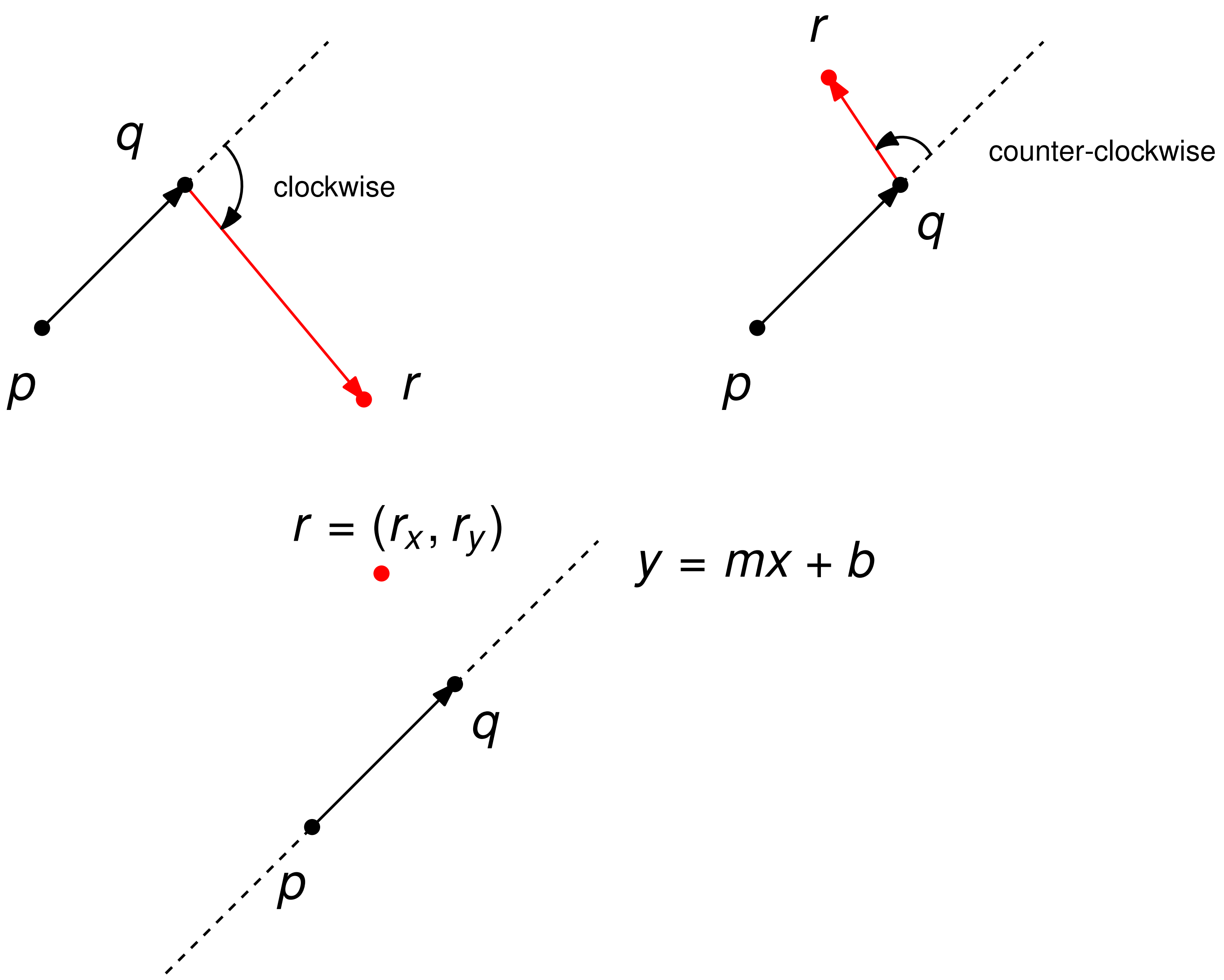
S is convex iff for every pair of points $p, q \in S$, the segment $\overline{pq}$ is in S.

Deciding the Ordering

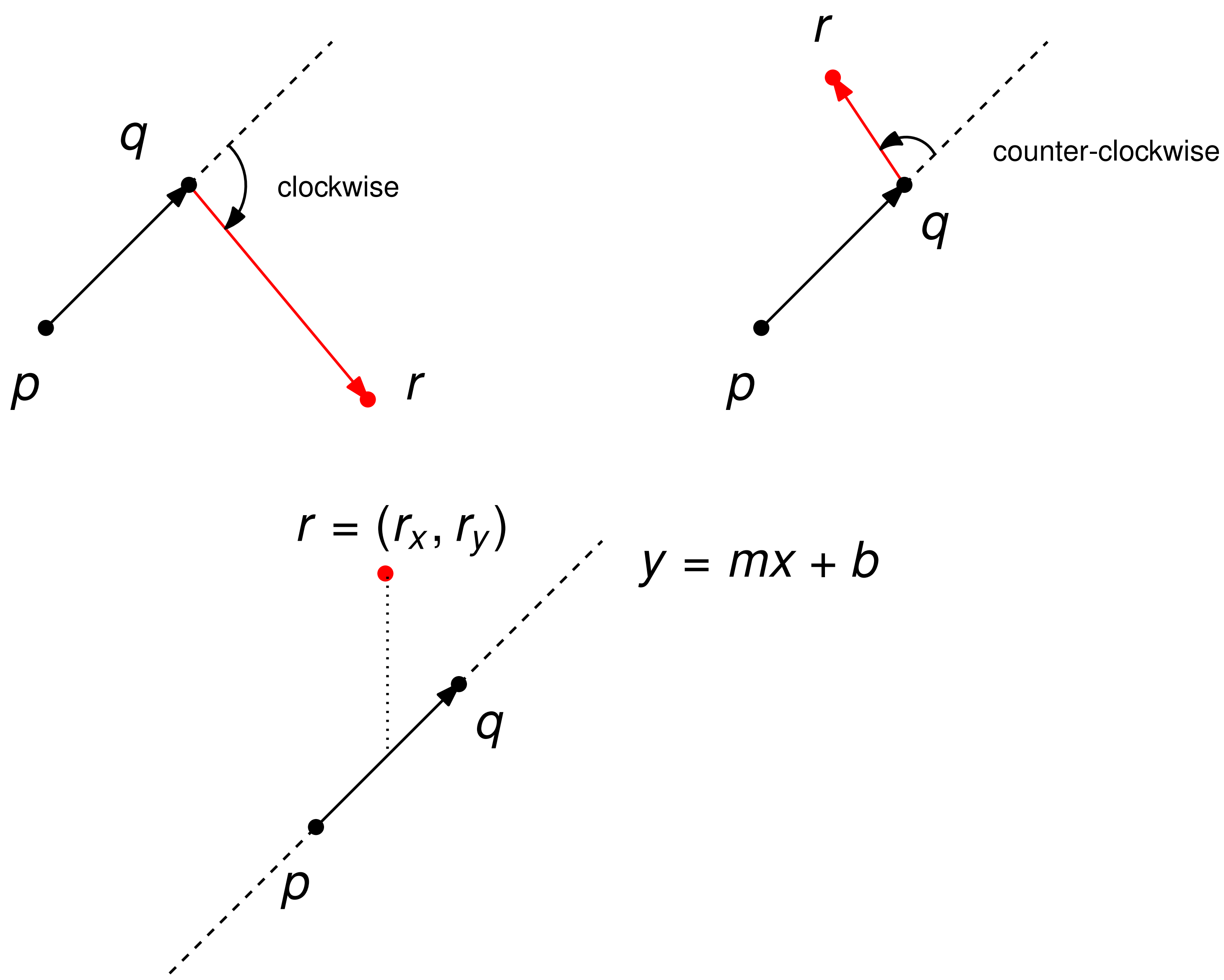




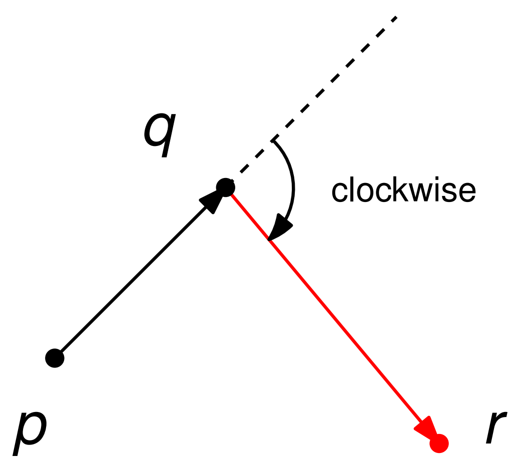
Deciding the Ordering

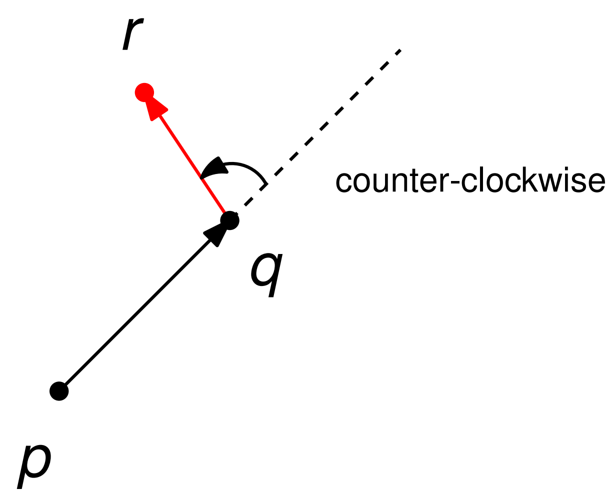


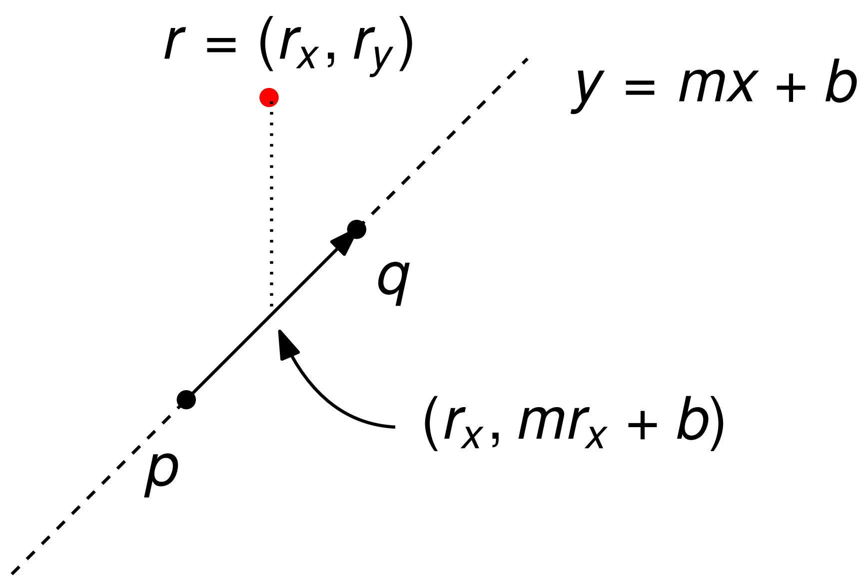
Deciding the Ordering



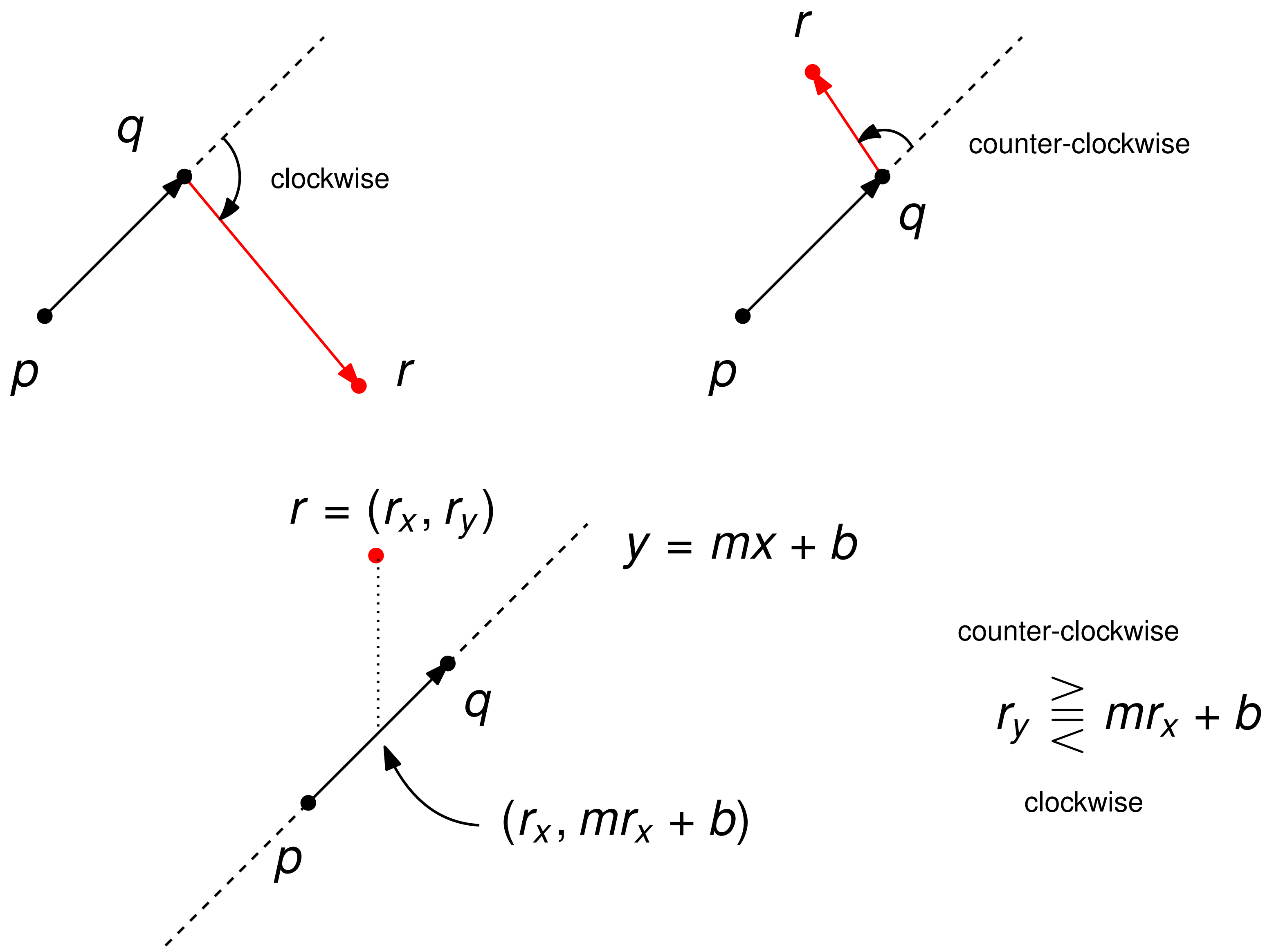
Deciding the Ordering







Deciding the Ordering

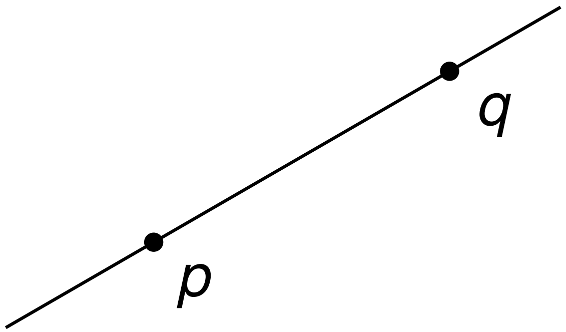


Line ℓ through two points p and q

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$

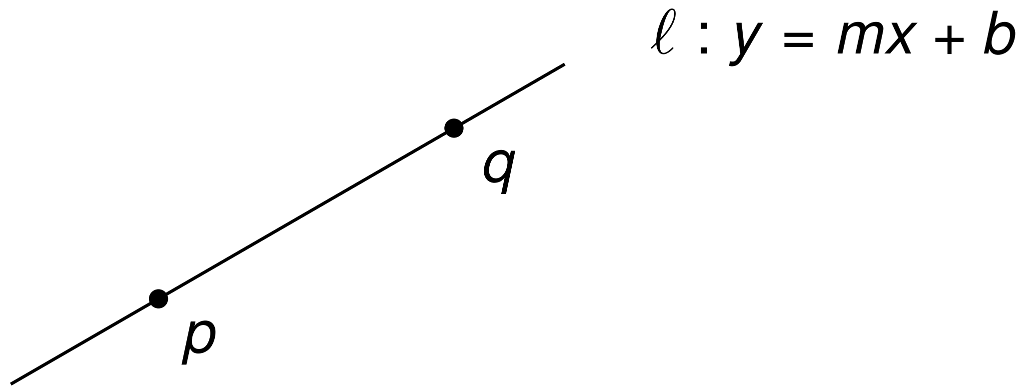
Line ℓ through two points p and q

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$



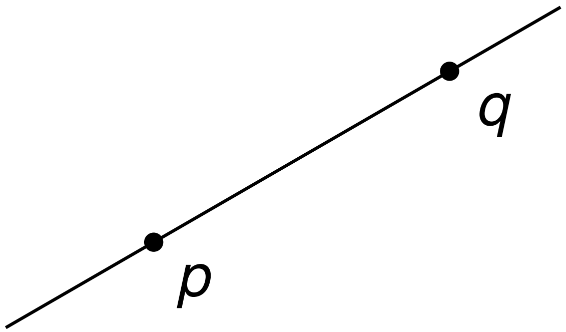
Line ℓ through two points p and q

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$



Line ℓ through two points p and q

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$



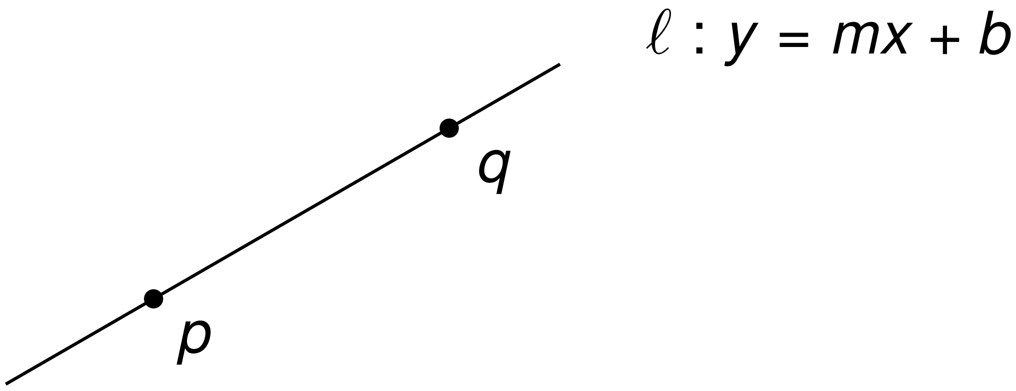
$\ell : y = mx + b$

$$\begin{cases} p_y = m \cdot p_x + b \\ q_y = m \cdot q_x + b \end{cases}$$

Solve for m and b

Line ℓ through two points p and q

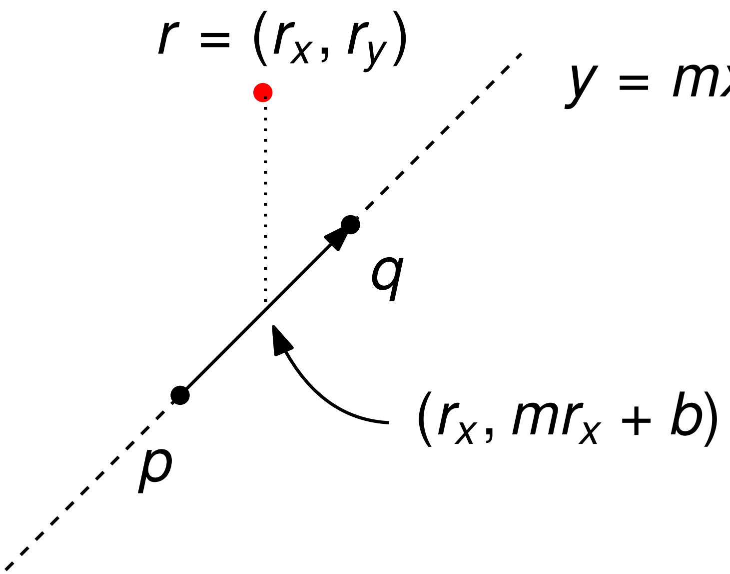
Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$



$$\begin{cases} p_y = m \cdot p_x + b \\ q_y = m \cdot q_x + b \end{cases}$$

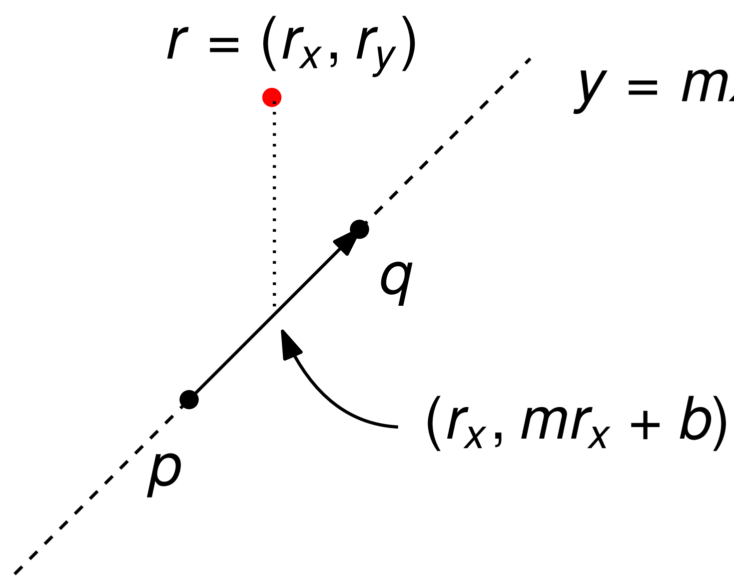
$$m = \frac{p_y - q_y}{p_x - q_x} \qquad b = \frac{p_x q_y - p_y q_x}{p_x - q_x}$$

Deciding the Ordering



$$y = mx + b = \frac{p_y - q_y}{p_x - q_x} \cdot x + \frac{p_x q_y - p_y q_x}{p_x - q_x}$$

Deciding the Ordering



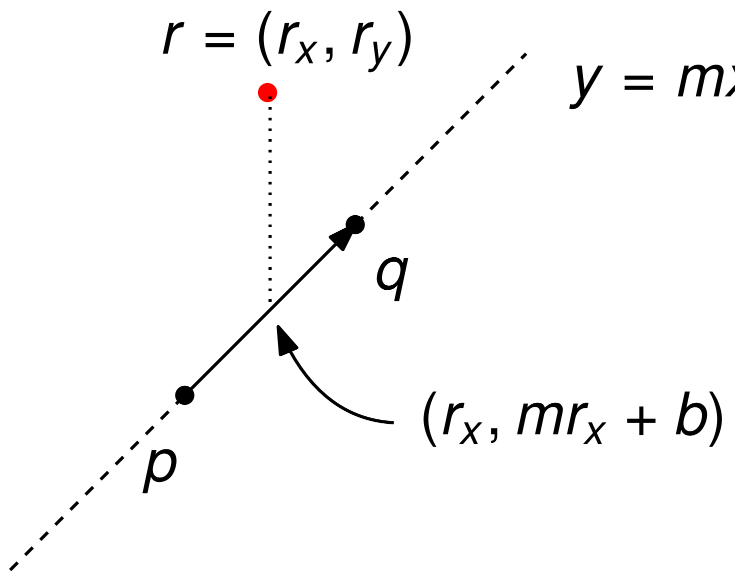
$$y = mx + b = \frac{p_y - q_y}{p_x - q_x} \cdot x + \frac{p_x q_y - p_y q_x}{p_x - q_x}$$

counter-clockwise

$$r_y \begin{matrix} > \\ = \\ < \end{matrix} mr_x + b = \frac{p_y - q_y}{p_x - q_x} \cdot r_x + \frac{p_x q_y - p_y q_x}{p_x - q_x}$$

clockwise

Deciding the Ordering



$$y = mx + b = \frac{p_y - q_y}{p_x - q_x} \cdot x + \frac{p_x q_y - p_y q_x}{p_x - q_x}$$

counter-clockwise

$$r_y \begin{array}{c} > \\ = \\ < \end{array} mr_x + b = \frac{p_y - q_y}{p_x - q_x} \cdot r_x + \frac{p_x q_y - p_y q_x}{p_x - q_x}$$

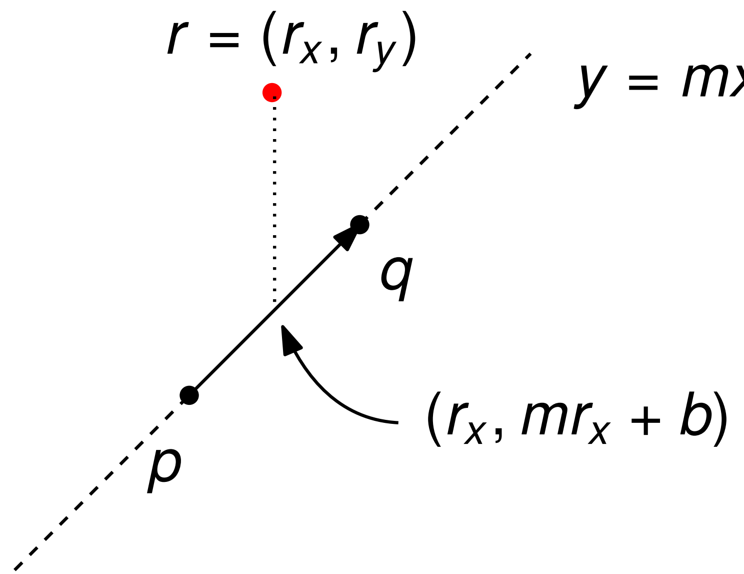
clockwise

counter-clockwise

$$(p_x - q_x) \cdot r_y \begin{array}{c} < \\ = \\ > \end{array} (p_y - q_y) r_x + (p_x q_y - p_y q_x)$$

clockwise

Deciding the Ordering



$$y = mx + b = \frac{p_y - q_y}{p_x - q_x} \cdot x + \frac{p_x q_y - p_y q_x}{p_x - q_x}$$

counter-clockwise

$$r_y \begin{array}{c} > \\ = \\ < \end{array} mr_x + b = \frac{p_y - q_y}{p_x - q_x} \cdot r_x + \frac{p_x q_y - p_y q_x}{p_x - q_x}$$

clockwise

change of sign, b/c

$p_x - q_x < 0$

counter-clockwise

$$(p_x - q_x) \cdot r_y \begin{array}{c} < \\ = \\ > \end{array} (p_y - q_y) r_x + (p_x q_y - p_y q_x)$$

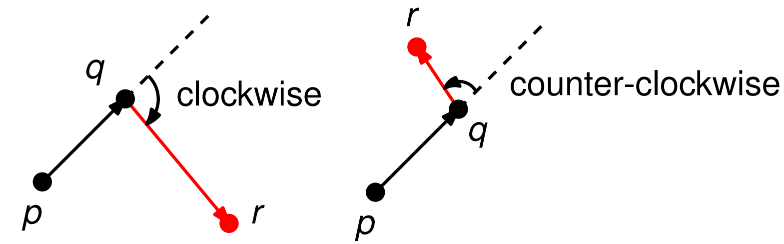
clockwise

Ordering via Matrix Determinant

counter-clockwise

$$\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} \begin{matrix} > \\ = \\ < \end{matrix} 0$$

clockwise

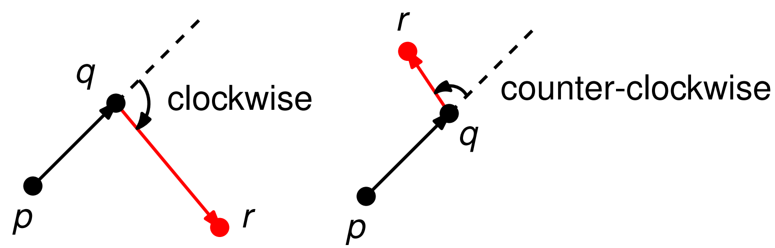


Ordering via Matrix Determinant

counter-clockwise

$$\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} \begin{matrix} > \\ = \\ < \end{matrix} 0$$

clockwise



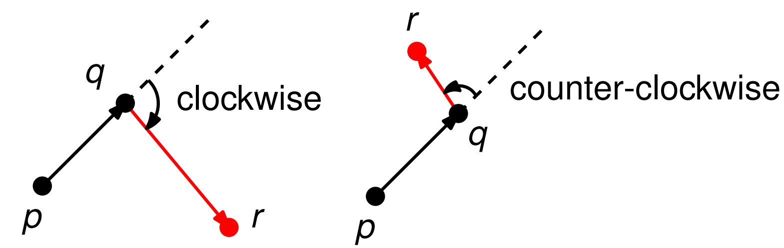
$$\begin{vmatrix} q_x & q_y \\ r_x & r_y \end{vmatrix} - p_x \cdot \begin{vmatrix} 1 & q_y \\ 1 & r_y \end{vmatrix} + p_y \cdot \begin{vmatrix} 1 & q_x \\ 1 & r_x \end{vmatrix} +$$

Ordering via Matrix Determinant

counter-clockwise

$$\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} \begin{matrix} > \\ = \\ < \end{matrix} 0$$

clockwise



$$\begin{vmatrix} q_x & q_y \\ r_x & r_y \end{vmatrix} - p_x \cdot \begin{vmatrix} 1 & q_y \\ 1 & r_y \end{vmatrix} + p_y \cdot \begin{vmatrix} 1 & q_x \\ 1 & r_x \end{vmatrix} +$$

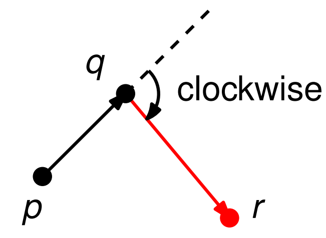
$$= (q_x r_y - r_x q_y) - p_x \cdot (r_y - q_y) + p_y \cdot (r_x - q_x)$$

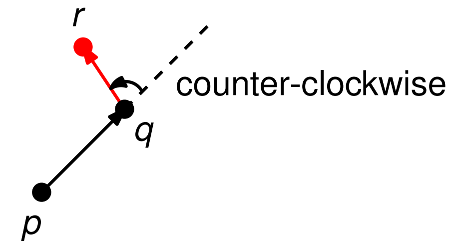
Ordering via Matrix Determinant

counter-clockwise

$$\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} \gtreqless 0$$

clockwise





$$\begin{vmatrix} q_x & q_y \\ r_x & r_y \end{vmatrix} - p_x \cdot \begin{vmatrix} 1 & q_y \\ 1 & r_y \end{vmatrix} + p_y \cdot \begin{vmatrix} 1 & q_x \\ 1 & r_x \end{vmatrix} +$$

$$= (q_x r_y - r_x q_y) - p_x \cdot (r_y - q_y) + p_y \cdot (r_x - q_x)$$

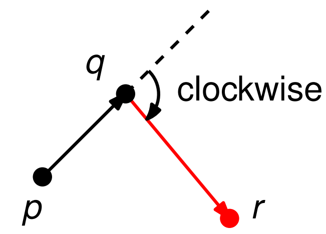
$$= q_x r_y - r_x q_y - p_x r_y + p_x q_y + p_y r_x - p_y q_x$$

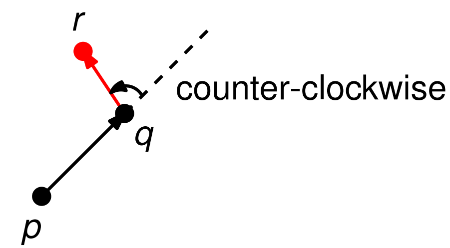
Ordering via Matrix Determinant

counter-clockwise

$$\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} \gtreqless 0$$

clockwise





$$\begin{vmatrix} q_x & q_y \\ r_x & r_y \end{vmatrix} - p_x \cdot \begin{vmatrix} 1 & q_y \\ 1 & r_y \end{vmatrix} + p_y \cdot \begin{vmatrix} 1 & q_x \\ 1 & r_x \end{vmatrix} +$$

$$= (q_x r_y - r_x q_y) - p_x \cdot (r_y - q_y) + p_y \cdot (r_x - q_x)$$

$$= \underline{q_x r_y} - \underline{r_x q_y} - \underline{p_x r_y} + p_x q_y + \underline{p_y r_x} - p_y q_x$$

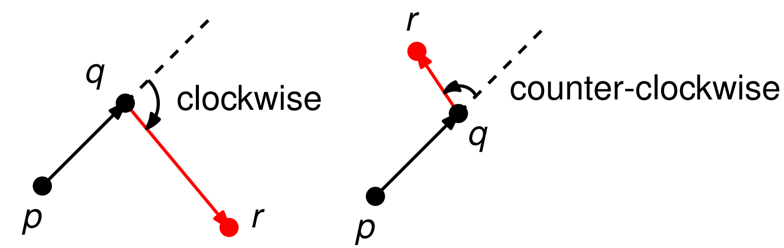
$$= \underline{-(p_x - q_x) \cdot r_y} + \underline{(p_y - q_y) \cdot r_x} + (p_x q_y - p_y q_x) \gtreqless 0$$

Ordering via Matrix Determinant

counter-clockwise

$$\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} \gtreqless 0$$

clockwise



$$\begin{vmatrix} q_x & q_y \\ r_x & r_y \end{vmatrix} - p_x \cdot \begin{vmatrix} 1 & q_y \\ 1 & r_y \end{vmatrix} + p_y \cdot \begin{vmatrix} 1 & q_x \\ 1 & r_x \end{vmatrix} +$$

$$= (q_x r_y - r_x q_y) - p_x \cdot (r_y - q_y) + p_y \cdot (r_x - q_x)$$

$$= \underline{q_x r_y} - \underline{r_x q_y} - \underline{p_x r_y} + p_x q_y + \underline{p_y r_x} - p_y q_x$$

$$= \underline{-(p_x - q_x) \cdot r_y} + \underline{(p_y - q_y) \cdot r_x} + (p_x q_y - p_y q_x) \gtreqless 0$$

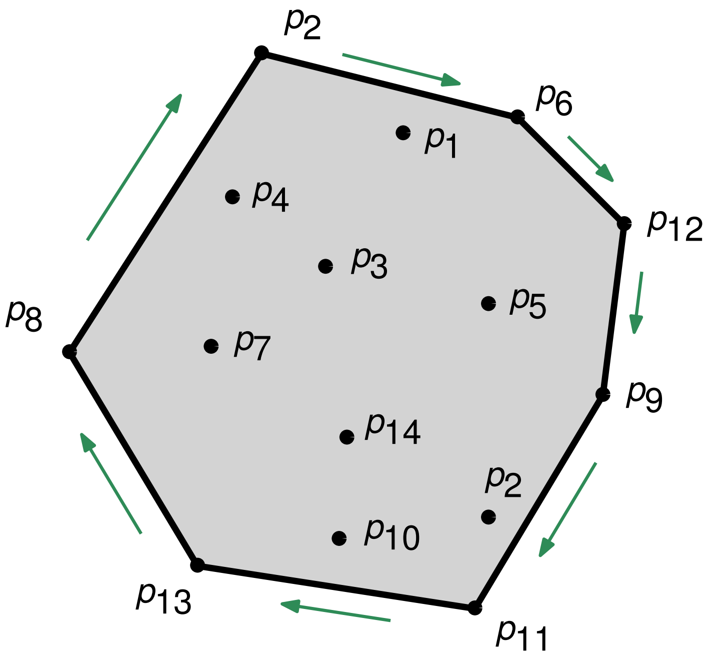
counter-clockwise

$$(p_y - q_y) \cdot r_x + (p_x q_y - p_y q_x) \gtreqless (p_x - q_x) \cdot r_y$$

clockwise

Convex Hull

- **Input:** Set of n points in $\mathbb{R}^2$
- **Output:** Smallest convex polygon that contains all points



Output:

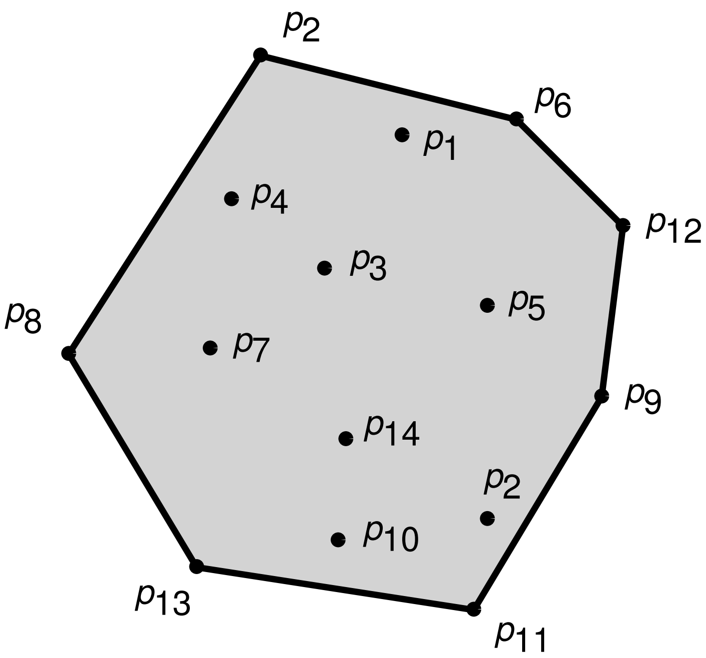
$p_2, p_6, p_{12}, p_9, p_{11}, p_{13}, p_8$

In clockwise order

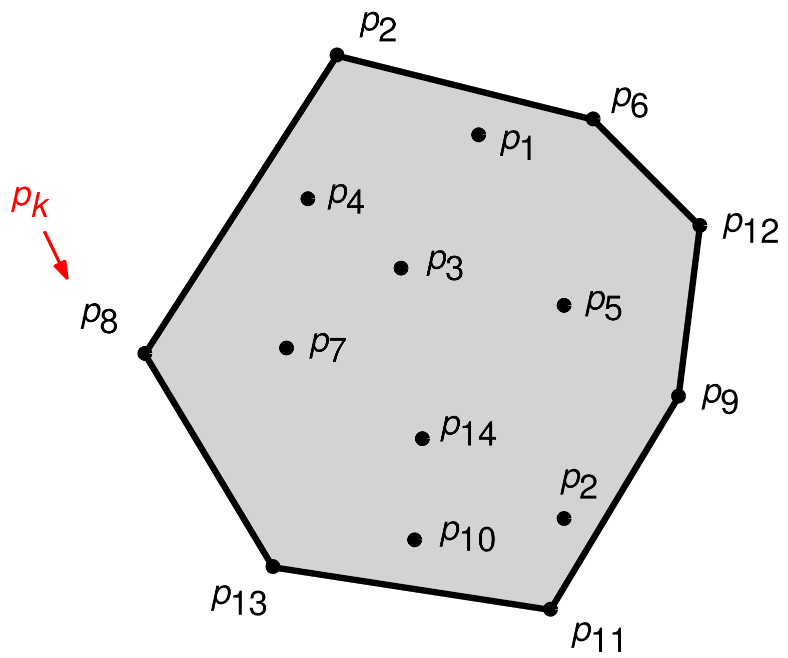
Polygon: circular sequence of points connected by line segments

S is convex iff for every pair of points $p, q \in S$, the segment $\overline{pq}$ is in S.

“Gift Wrapping” Algorithm

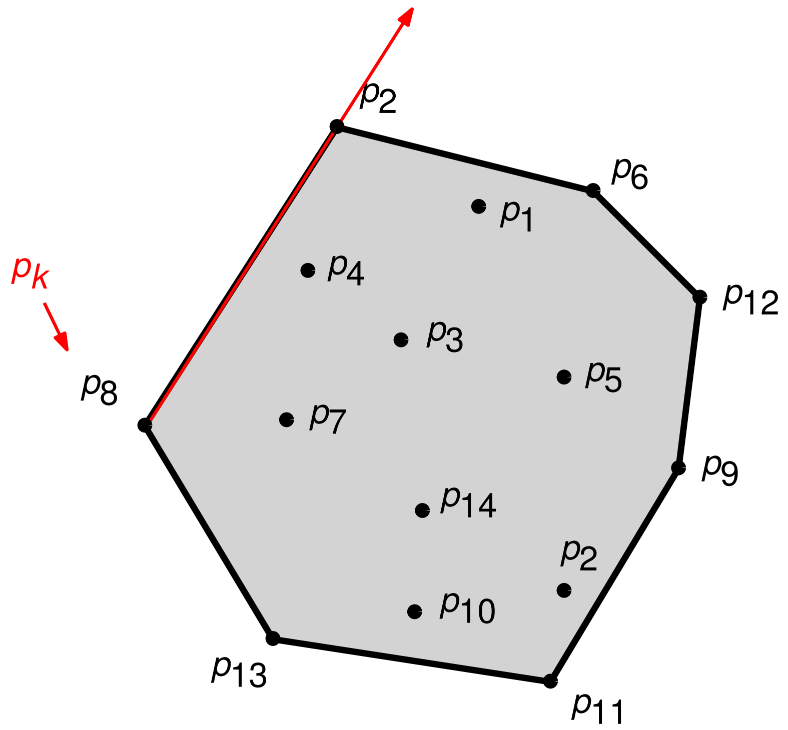


“Gift Wrapping” Algorithm



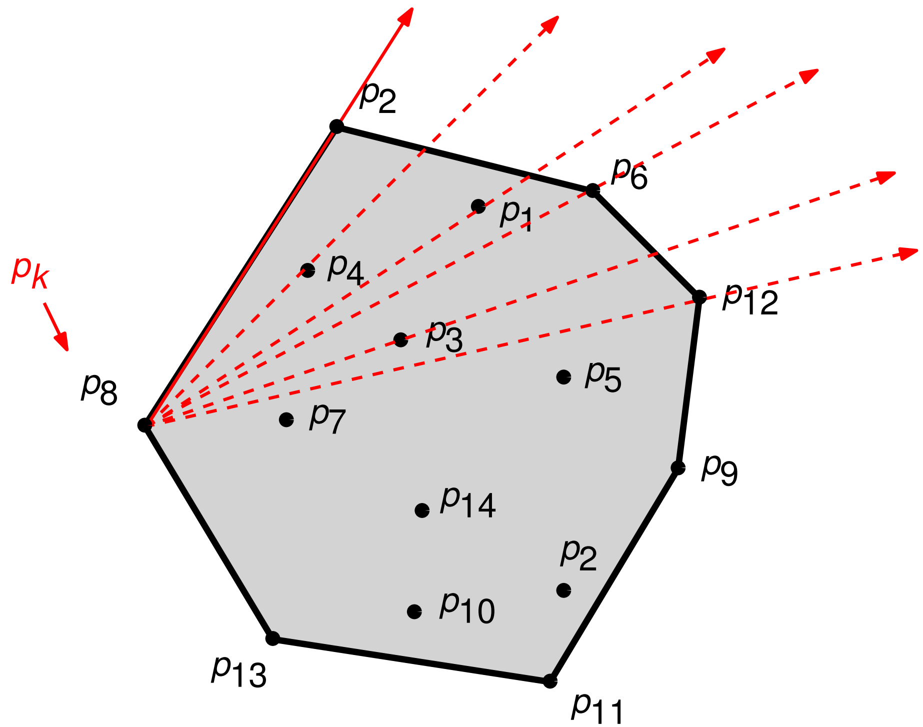
Let p_k be the leftmost point

"Gift Wrapping" Algorithm



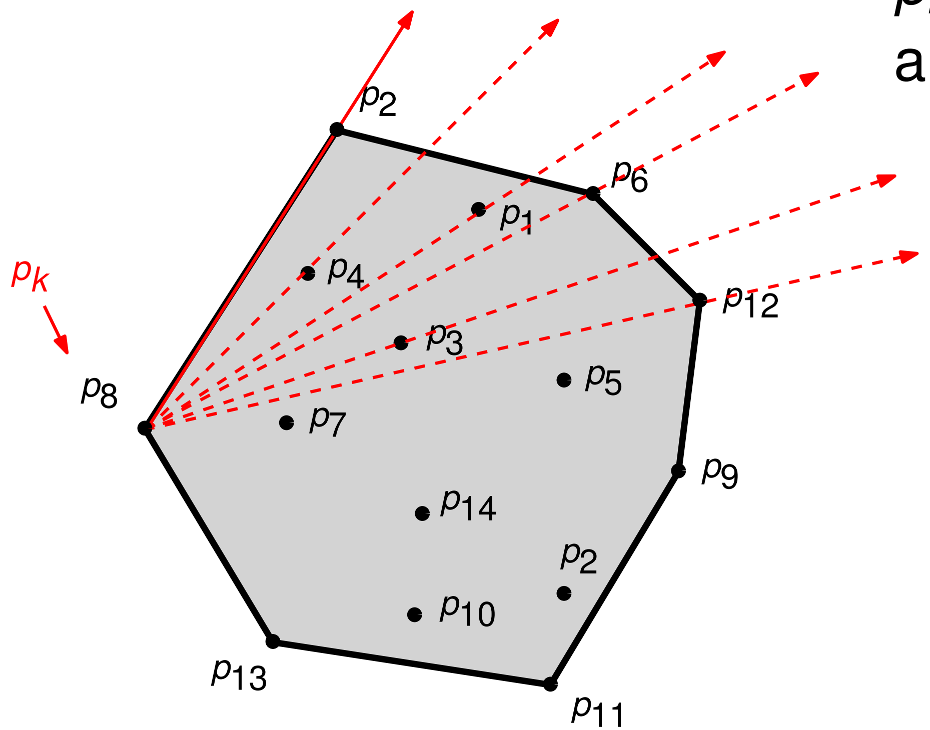
Let p_k be the leftmost point

“Gift Wrapping” Algorithm



Let p_k be the leftmost point

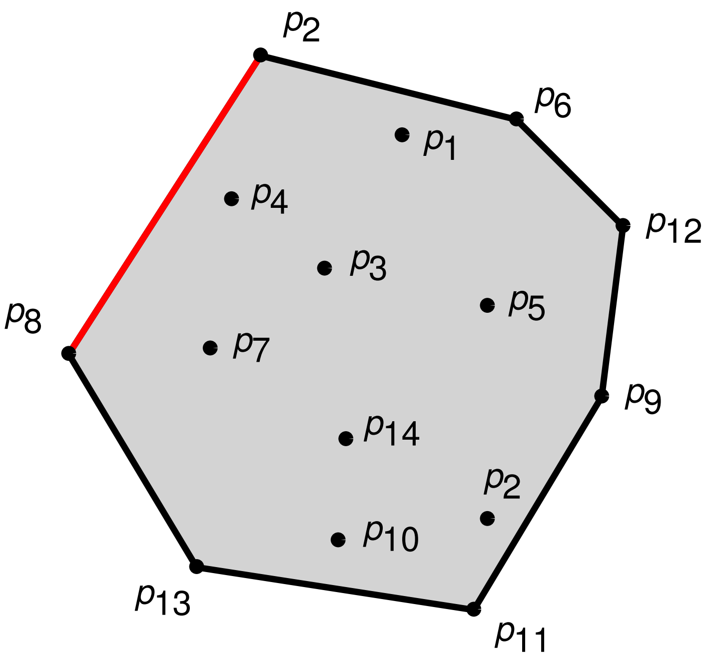
"Gift Wrapping" Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

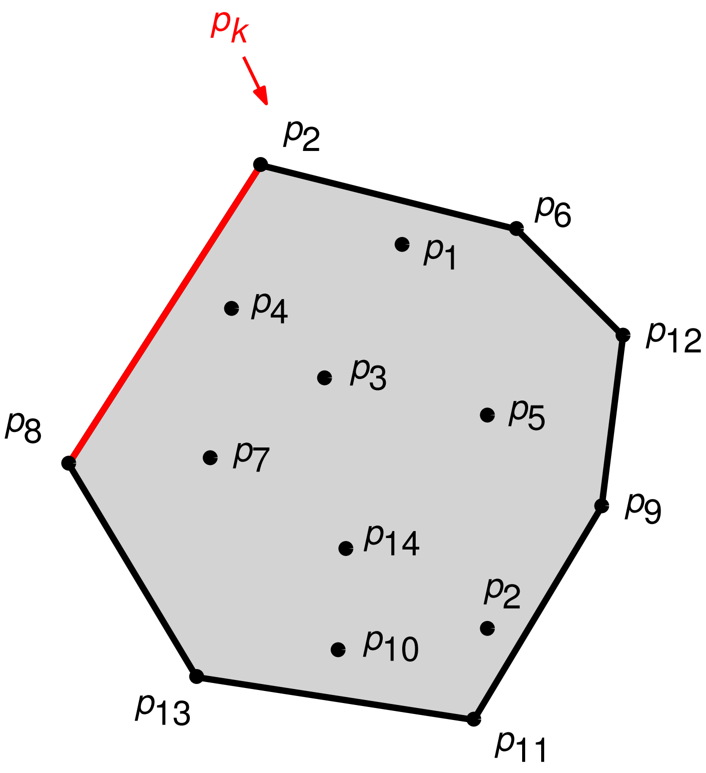
“Gift Wrapping” Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

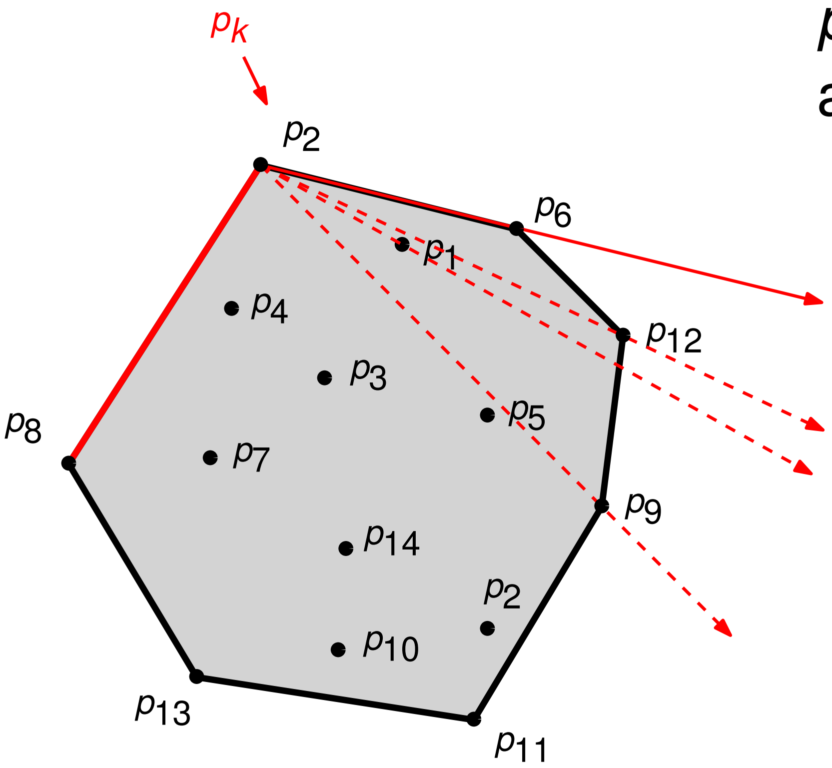
"Gift Wrapping" Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

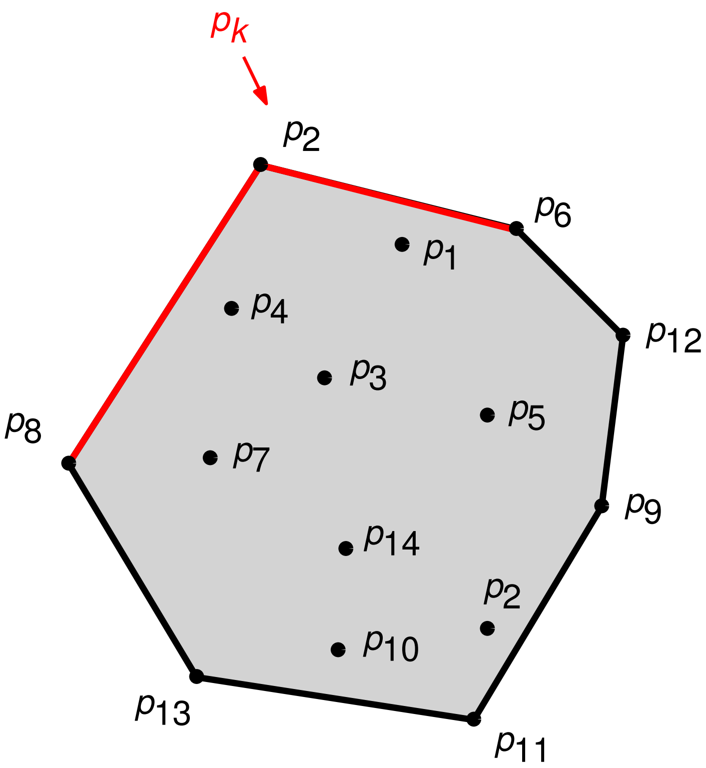
“Gift Wrapping” Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

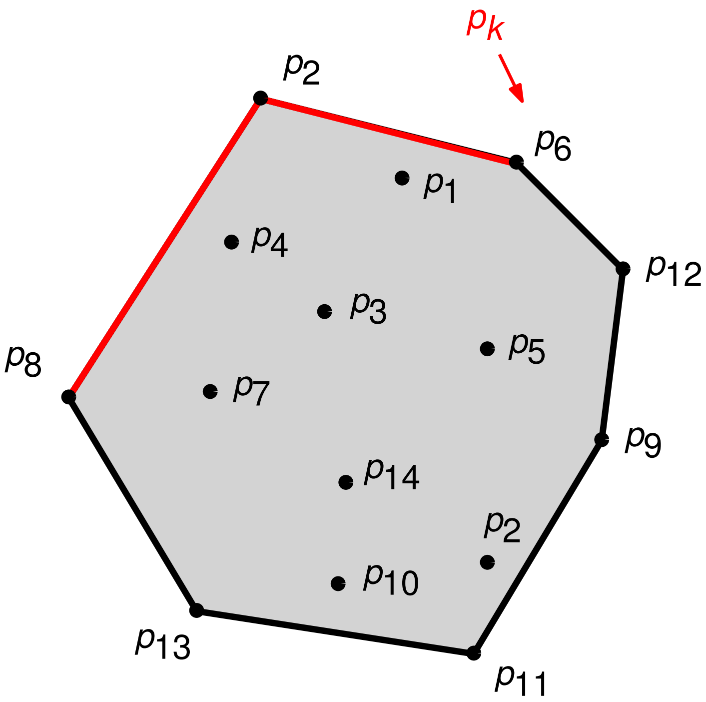
“Gift Wrapping” Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

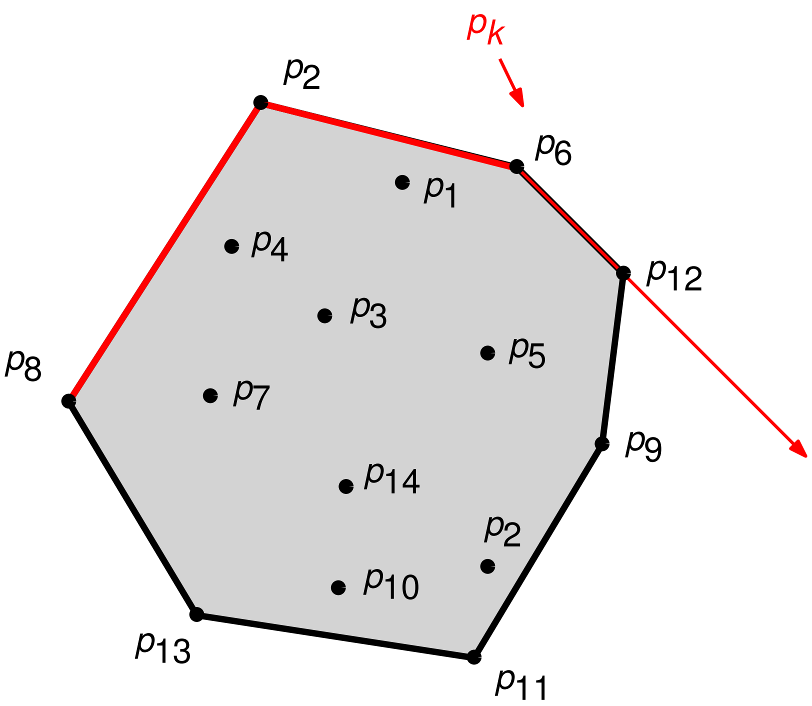
"Gift Wrapping" Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

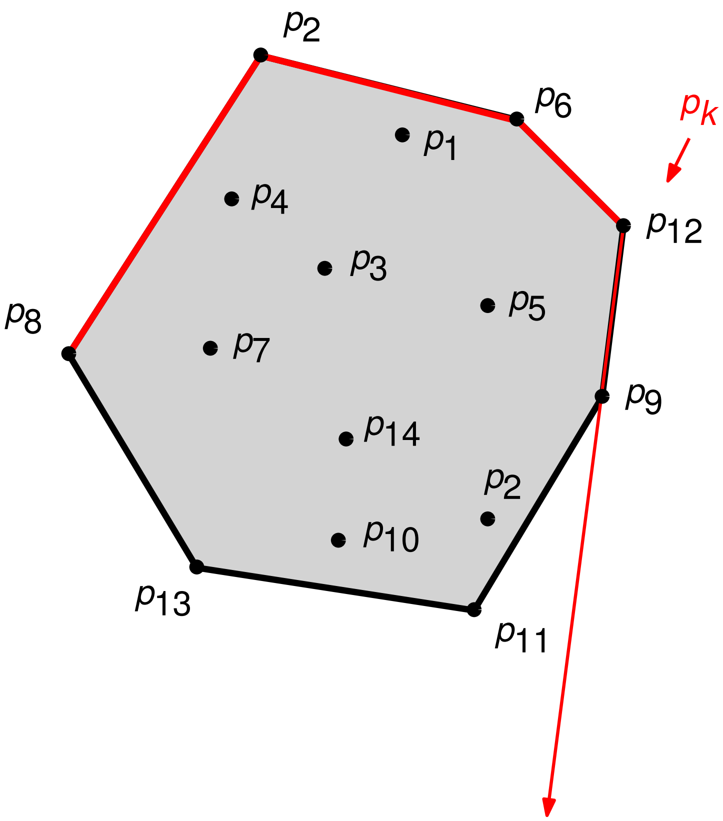
“Gift Wrapping” Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

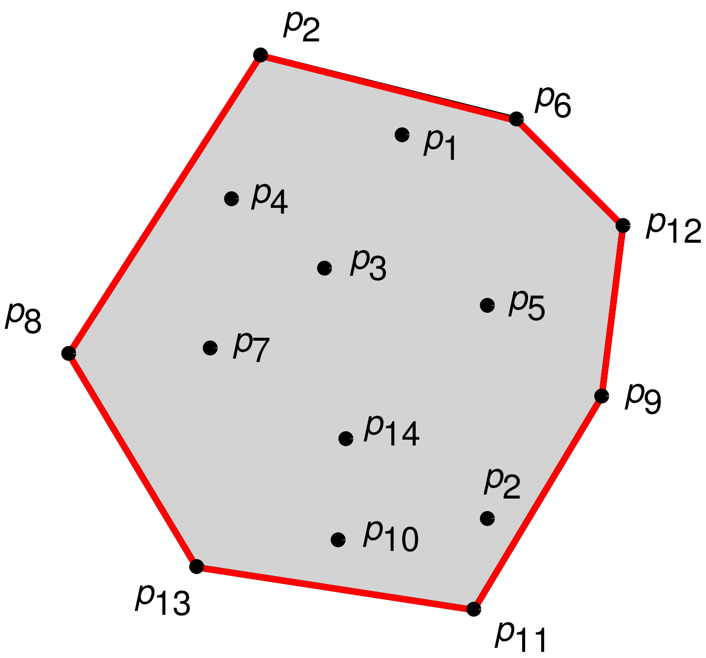
"Gift Wrapping" Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

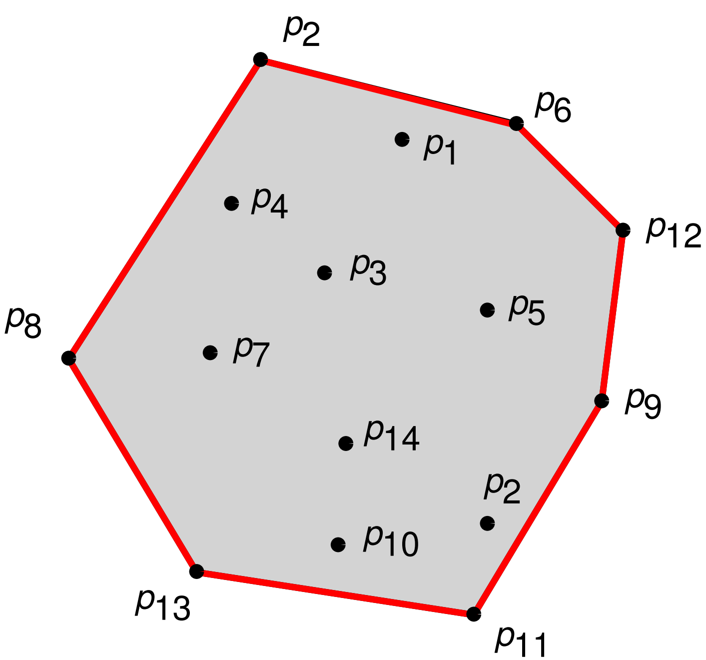
"Gift Wrapping" Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

“Gift Wrapping” Algorithm

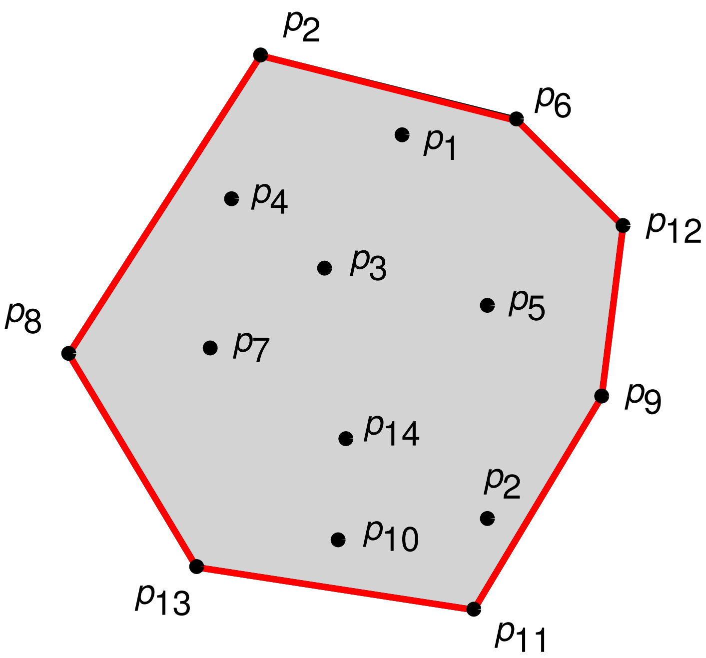


Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

procedure $\textsc{GiftWrap}([p_1..p_n])$
    $s = \arg\min_x p_i$
    $k = s$
    repeat
        $j = \textsc{findMaxSlope}(p_k, [p_1..p_n])$
        $\textsc{print}(p_j)$
        $k = j$
    until k = s
end procedure

"Gift Wrapping" Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

procedure GIFTWRAP($[p_1..p_n]$)

$O(n)$ $\quad s = \arg\min_x p_i$

$O(1)$ $\quad k = s$

repeat

$O(n)$ $\quad j =$ FINDMAXSLOPE($p_k, [p_1..p_n]$)

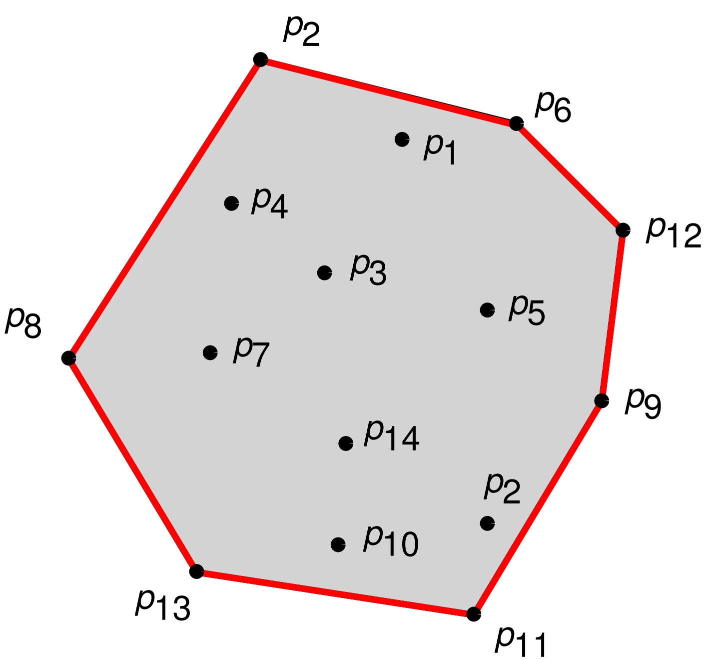
$O(1)$ $\quad$ PRINT(p_j)

$O(1)$ $\quad k = j$

until k = s

end procedure

"Gift Wrapping" Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

procedure GIFTWRAP($[p_1..p_n]$)

$O(n)$ $s = \arg\min_x p_i$

$O(1)$ $k = s$

repeat

? $O(n)$ $j =$ FINDMAXSLOPE($p_k, [p_1..p_n]$)

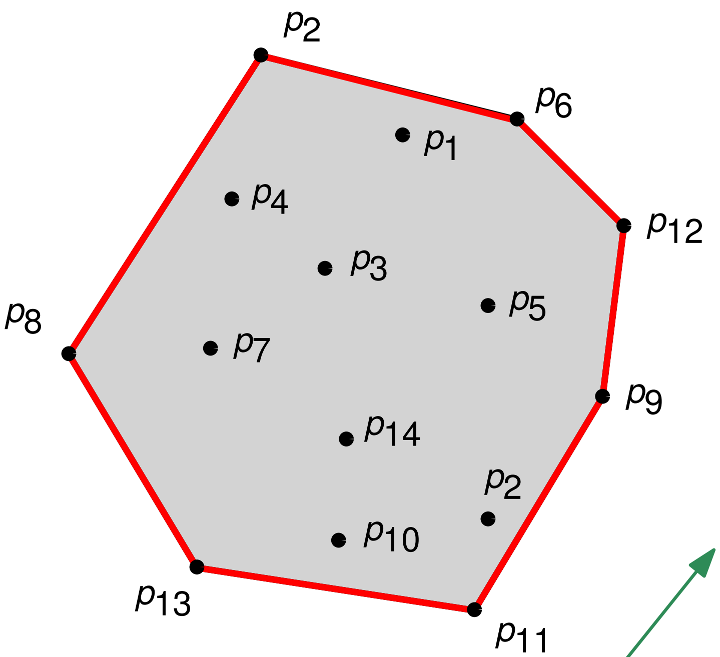
$O(1)$ PRINT(p_j)

$O(1)$ $k = j$

until k = s

end procedure

"Gift Wrapping" Algorithm



h: # of points on the hull

Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

procedure GIFTWRAP($[p_1..p_n]$)

$O(n)$ $\quad s = \arg\min_x p_i$

$O(1)$ $\quad k = s$

repeat

? [

$O(n)$ $\quad j = \text{FINDMAXSLOPE}(p_k, [p_1..p_n])$

$O(1)$ $\quad \text{PRINT}(p_j)$

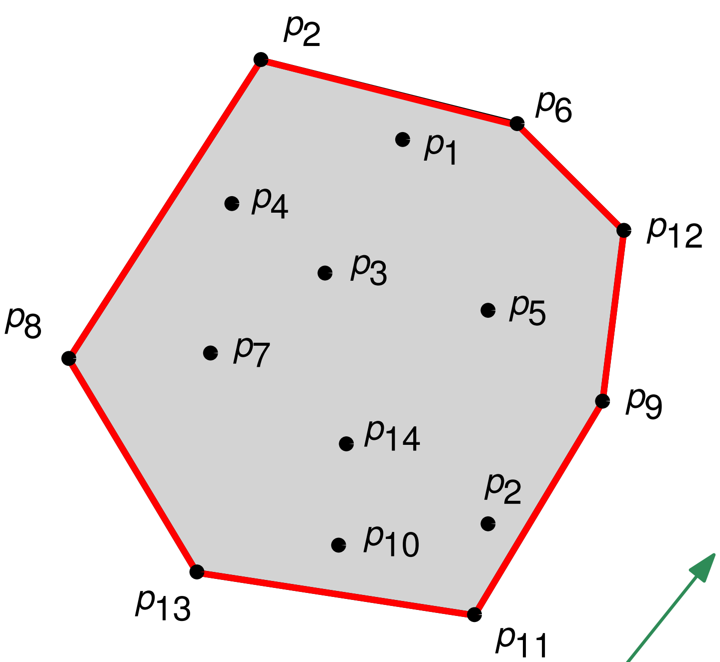
$O(1)$ $\quad k = j$

]

until k = s

end procedure

“Gift Wrapping” Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

procedure GIFTWRAP($[p_1..p_n]$)

$O(n)$ $s = \arg\min_x p_i$

$O(1)$ $k = s$

repeat

?

$O(n)$ $j = \text{FINDMAXSLOPE}(p_k, [p_1..p_n])$

$O(1)$ PRINT(p_j)

$O(1)$ $k = j$

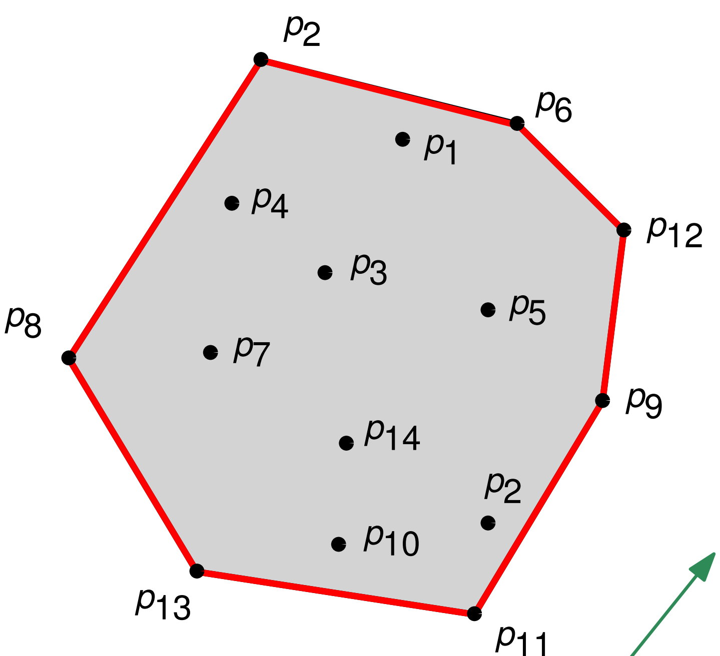
until k = s

end procedure

h: # of points on the hull

$O(nh)$

“Gift Wrapping” Algorithm



Let p_k be the leftmost point

$\overline{p_k, p_2}$ has the largest slope among all $\overline{p_k, p_i}$

procedure GIFTWRAP($[p_1..p_n]$)

$O(n)$ $s = \arg\min_x p_i$

$O(1)$ $k = s$

repeat

?

$O(n)$ $j =$ FINDMAXSLOPE($p_k, [p_1..p_n]$)

$O(1)$ PRINT(p_j)

$O(1)$ $k = j$

until k = s

end procedure

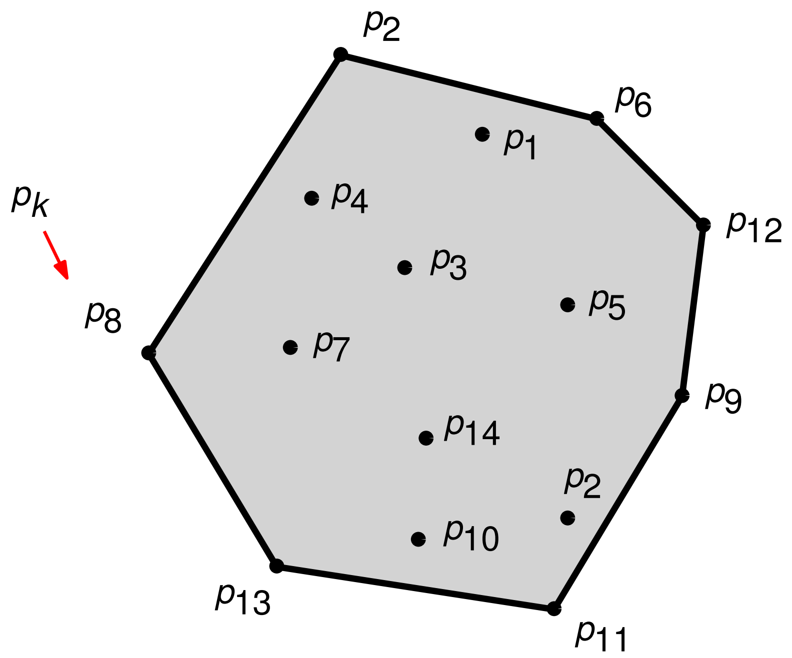
h: # of points on the hull

$O(nh)$

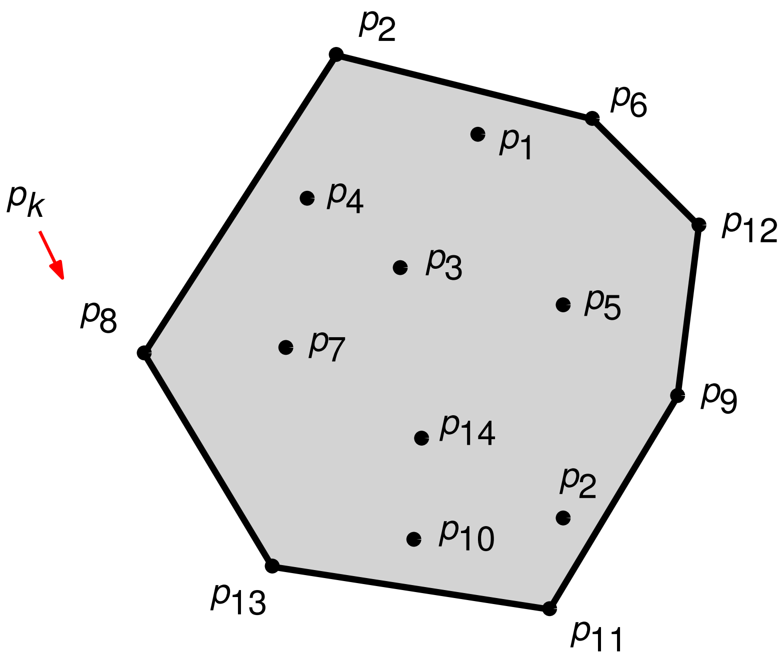
If $h = n$, then $O(n^2)$

Graham's Scan

Let p_k be the leftmost point



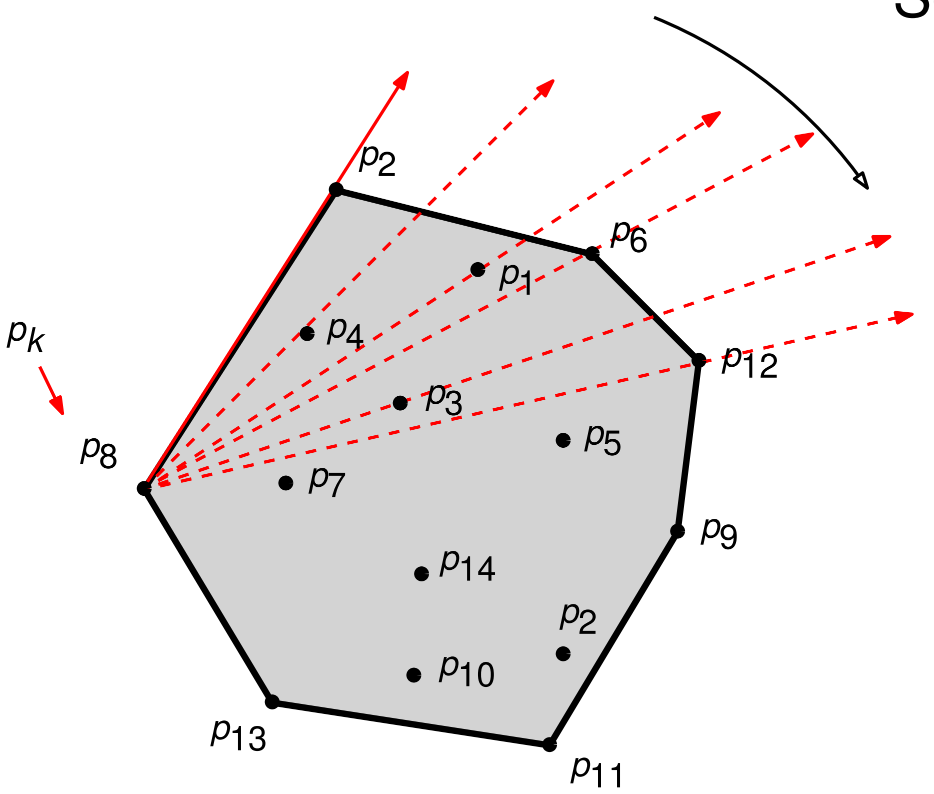
Graham's Scan



Let p_k be the leftmost point

Sort all points radially around p_k

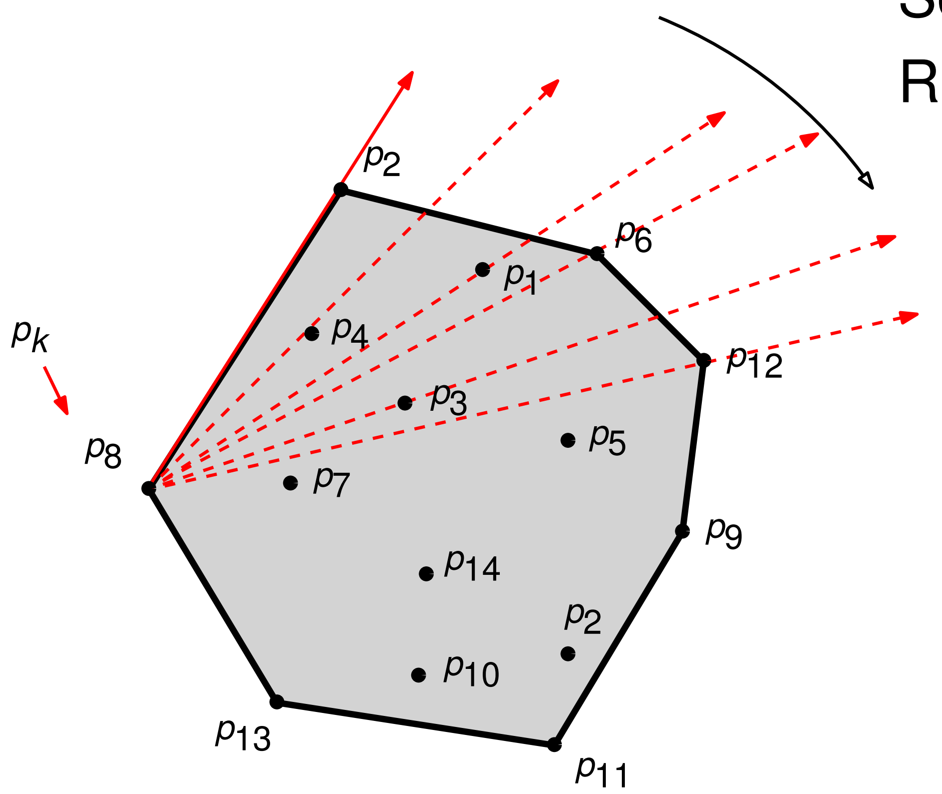
Graham's Scan



Let p_k be the leftmost point

Sort all points radially around p_k

Graham's Scan

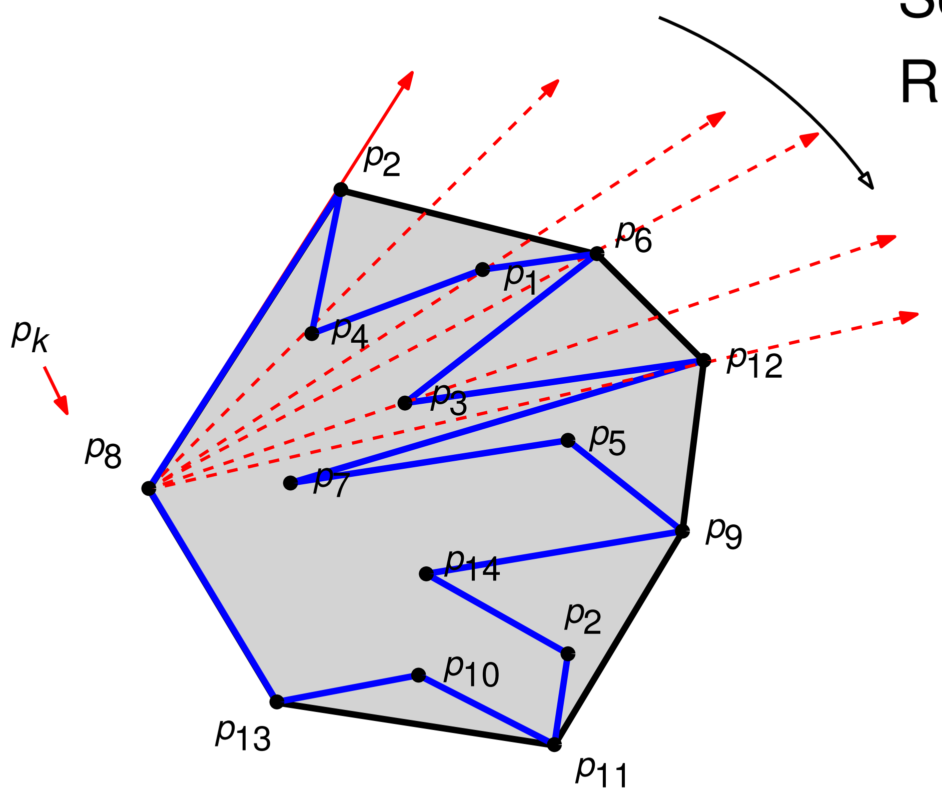


Let p_k be the leftmost point

Sort all points radially around p_k

Report points in that order

Graham's Scan



Let p_k be the leftmost point

Sort all points radially around p_k

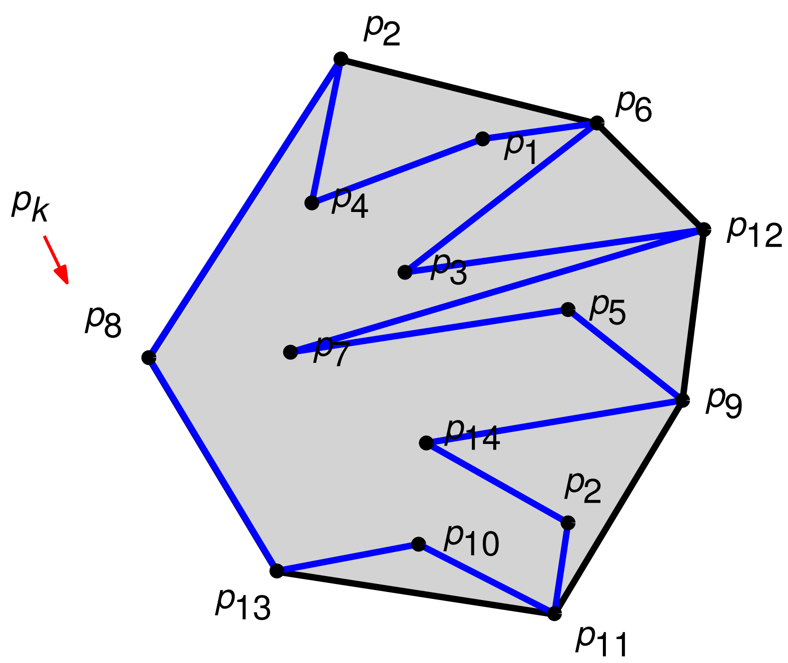
Report points in that order

Graham's Scan

Let p_k be the leftmost point

Sort all points radially around p_k

Report points in that order



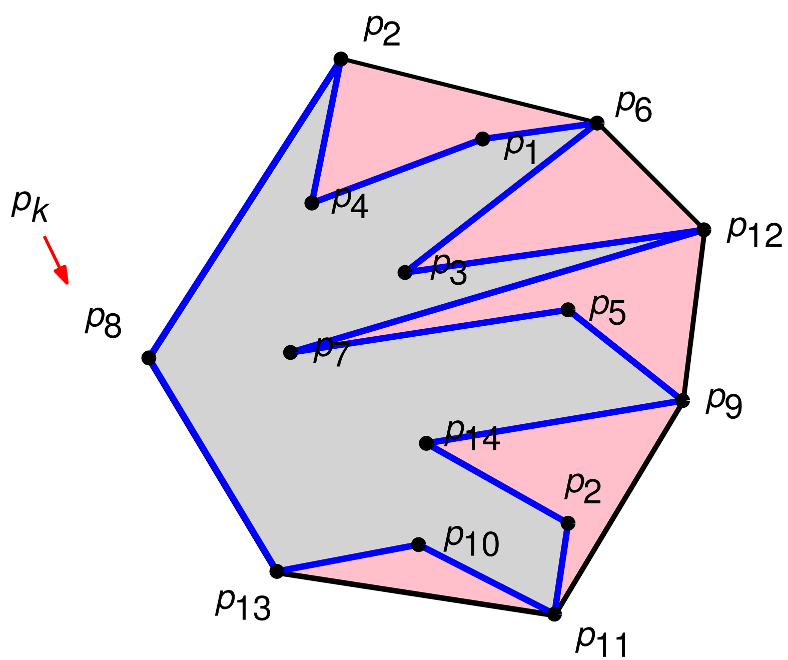
Simple polygon, but not convex

Graham's Scan

Let p_k be the leftmost point

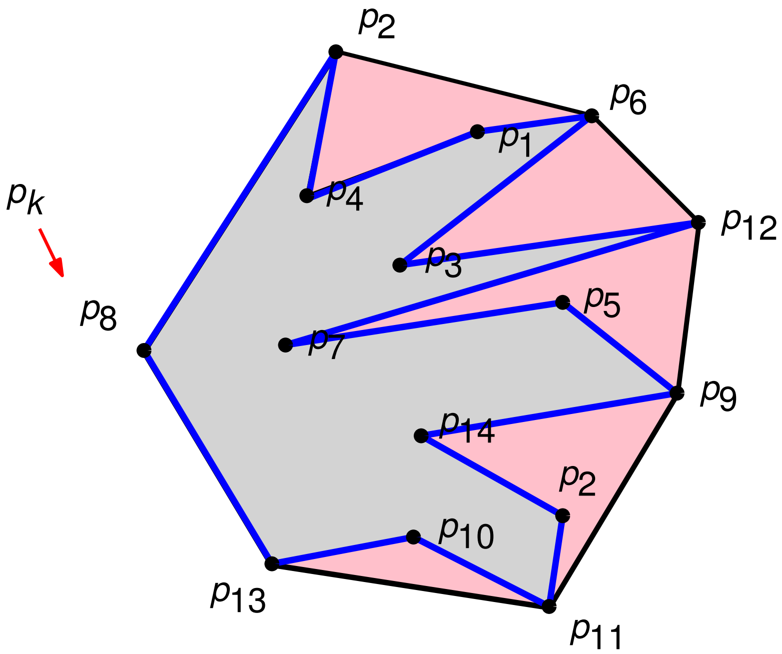
Sort all points radially around p_k

Report points in that order

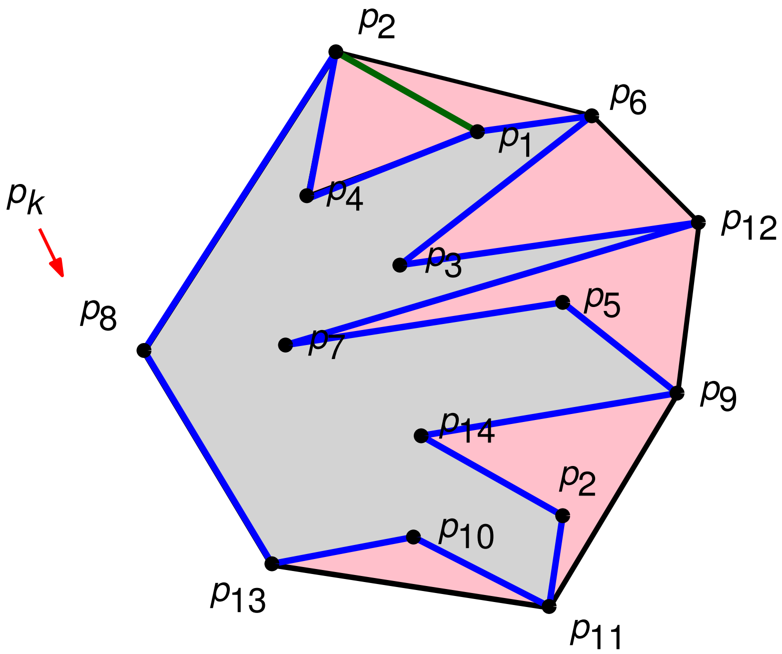


Simple polygon, but not convex

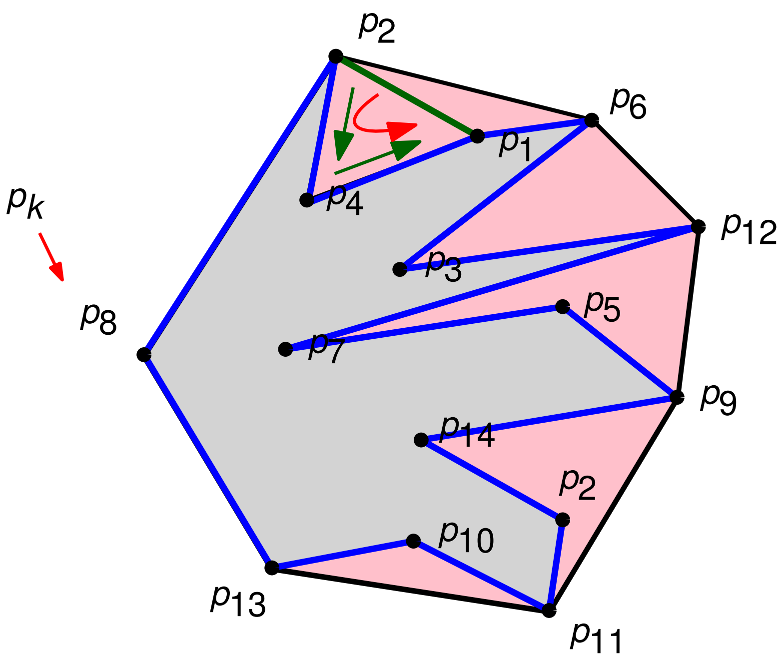
Fixing Graham's Scan



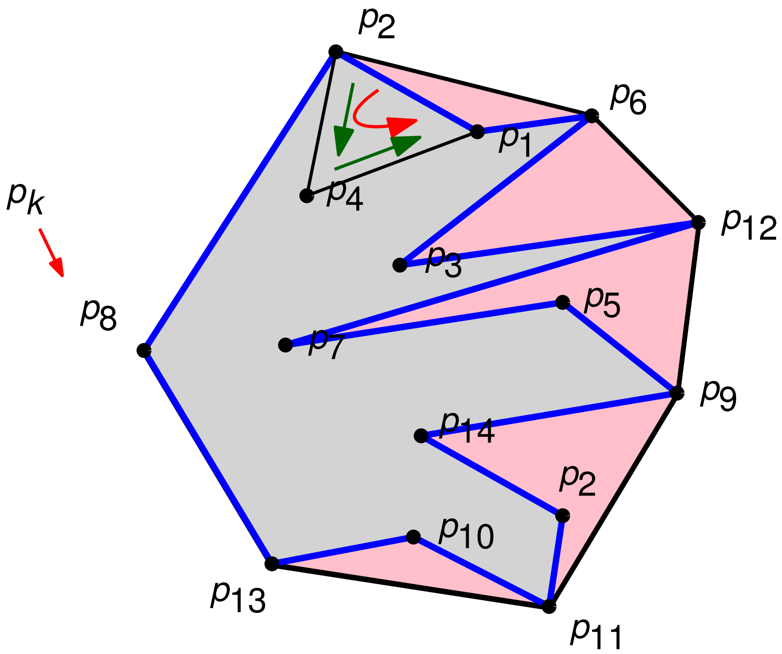
Fixing Graham's Scan



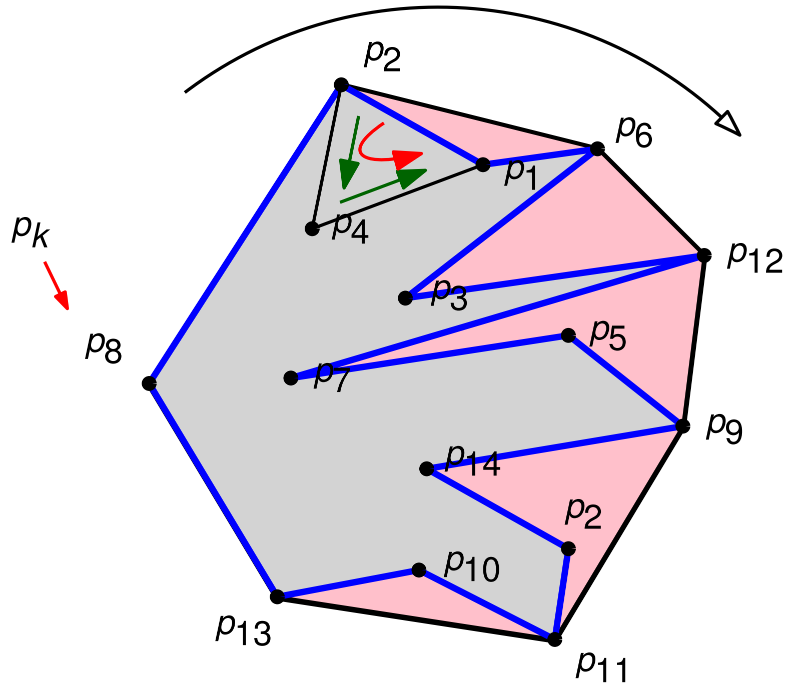
Fixing Graham's Scan



Fixing Graham's Scan

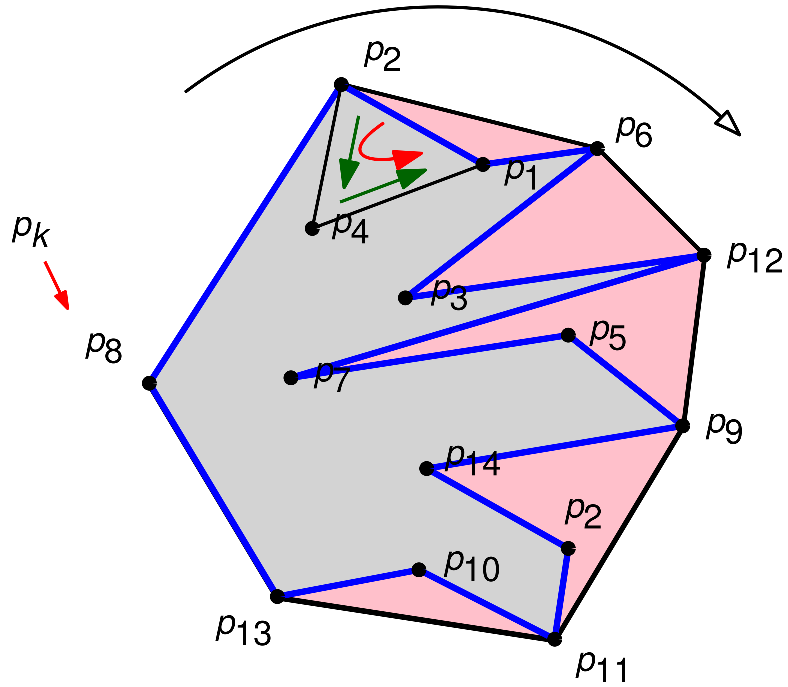


Fixing Graham's Scan



- Identify a reflex vertex in $O(1)$ time
- Eliminate reflex vertex in $O(1)$ time
- Repeat until no reflex vertices remain

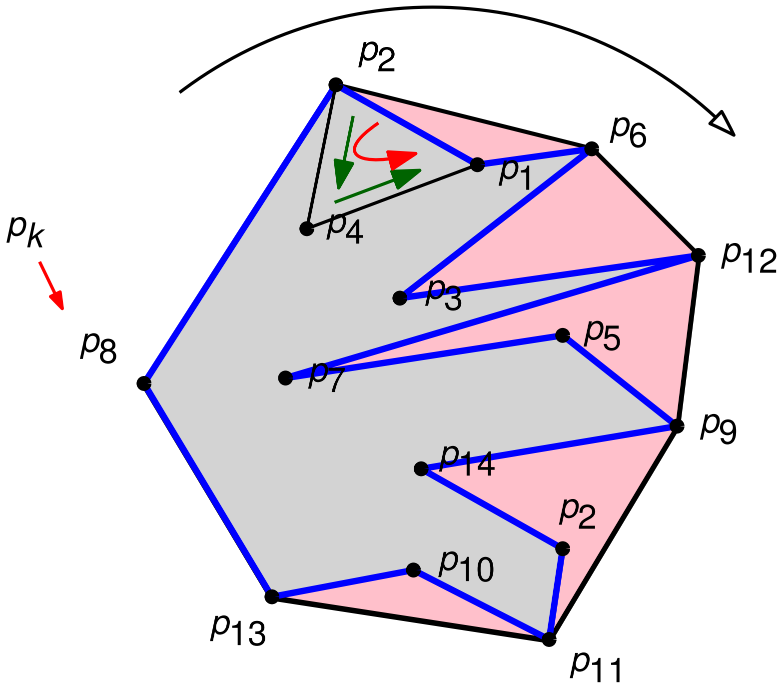
Fixing Graham's Scan



- Identify a reflex vertex in $O(1)$ time
- Eliminate reflex vertex in $O(1)$ time
- Repeat until no reflex vertices remain

$O(n)$ time

Fixing Graham's Scan



- Identify a reflex vertex in $O(1)$ time
- Eliminate reflex vertex in $O(1)$ time
- Repeat until no reflex vertices remain

$O(n)$ time

$O(n \log n)$ time for initial sorting