Departamento de Sistemas e Computação – FURB
Curso de Ciência da Computação
Disciplina de Processamento de imagens

Envoltória convexa

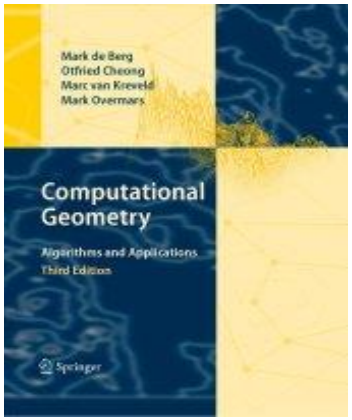
Prof. Aurélio Hoppe

aureliof@furb.br

http://www.inf.furb.br/~aurelio/

Grupo de Processamento de Imagens,
Análise de dados, Robótica e
Simulação computacional

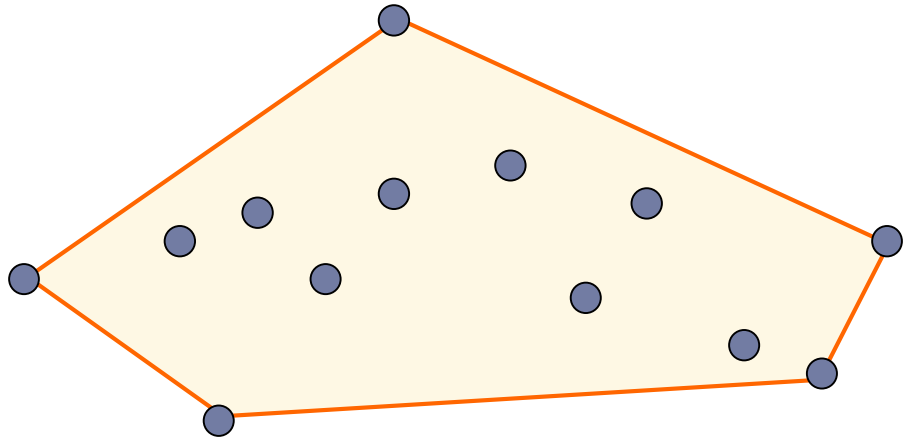
Bibliografia



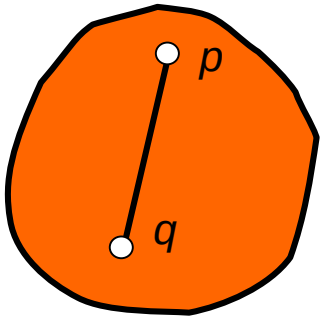
Computational geometry: algorithms and applications
Mark Berg
New York : Springer, 2000, 367p.

ENVOLTÓRIA CONVEXA

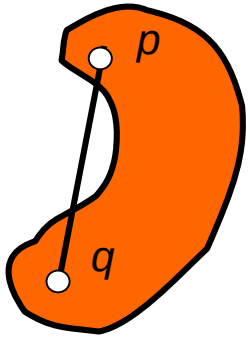
- **Enunciado:** Dados n pontos queremos encontrar os pontos extremos.



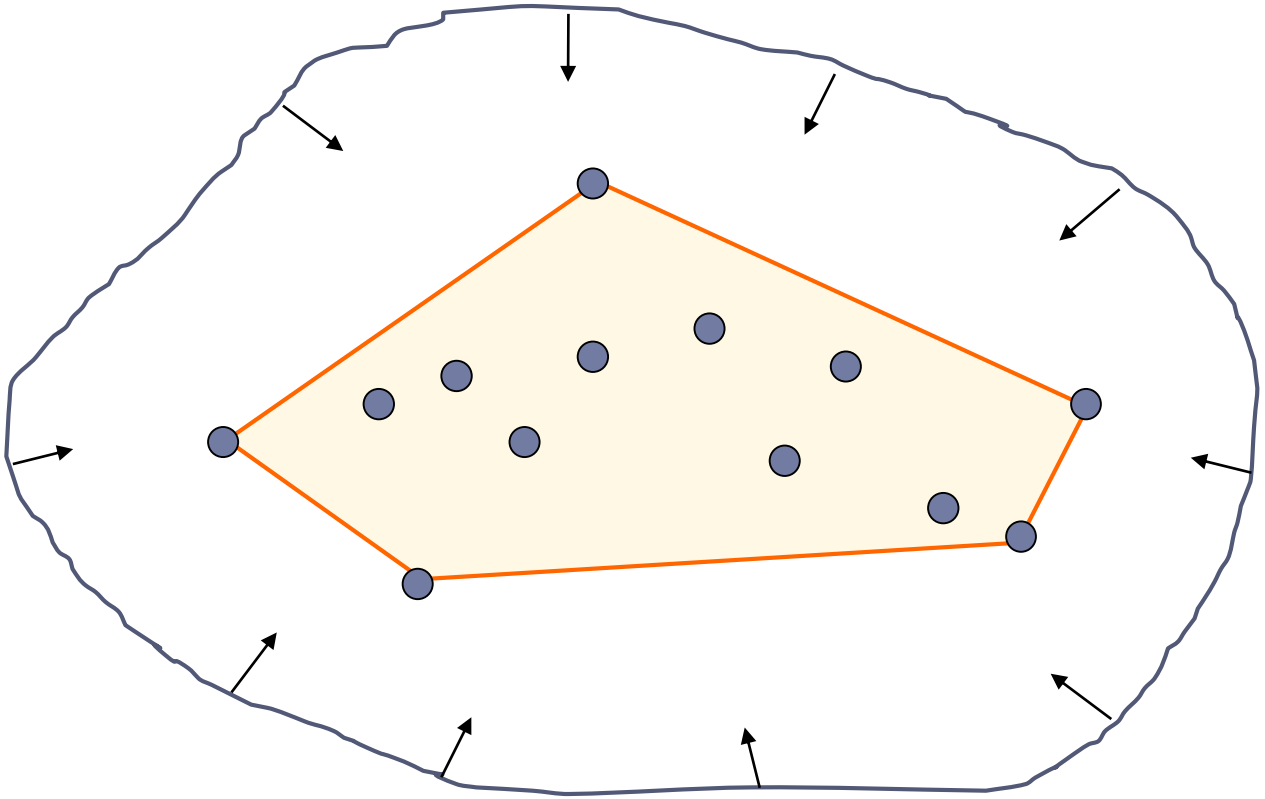
ENVOLTÓRIA CONVEXA



convexo

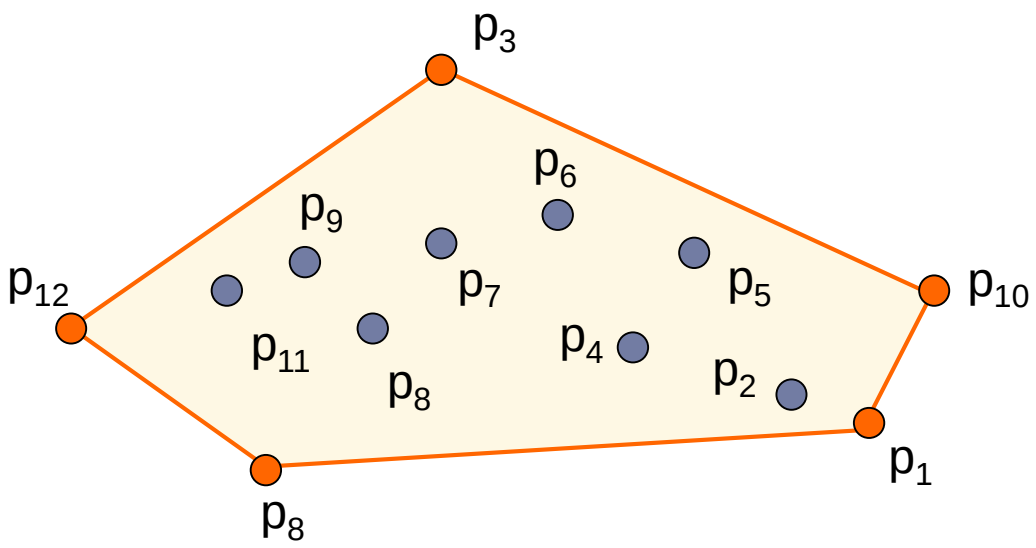


não convexo



ENVOLTÓRIA CONVEXA

- **Entrada:** p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12



EC é um polígono convexo

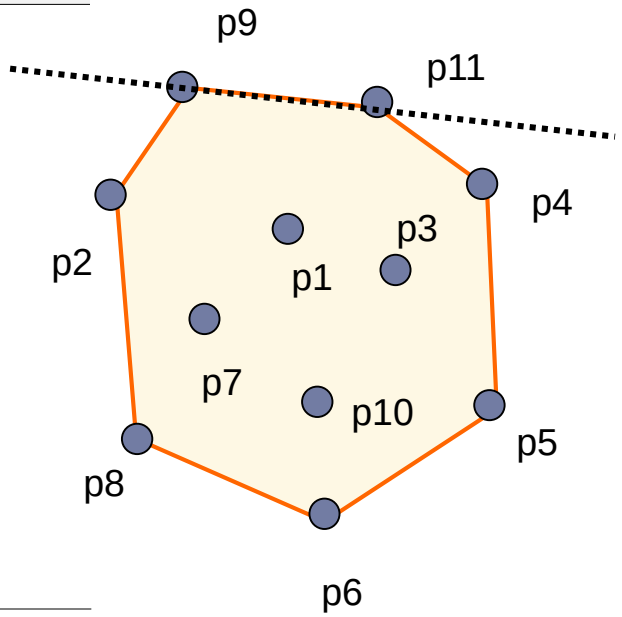
- **Saída:** p12, p8, p3, p1, p10

ENVOLTÓRIA CONVEXA
[Algoritmo ingênuo]

- **Entrada:** Conjunto de pontos P no plano

Algoritmo ingênuo (EC1)

```
01.  E ⇦ ∞
02.  FOR todos pares (p,q) ∈ PxP (p!=q)
03.      DO valido = true
04.      FOR todos pontos r ∈ P ( r!=p and r!=q )
05.          IF r está a direita da reta pq
06.          THEN valido = false
07.      IF valido THEN adicione pq para E
08.  Devolve conjunto E de arestas
```

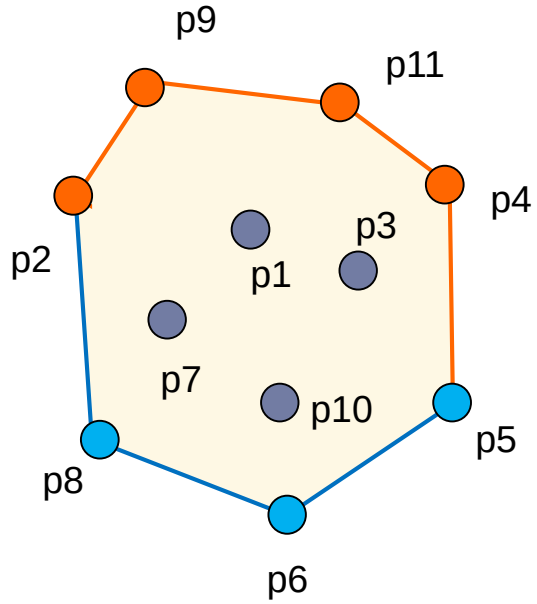


- **Saída:** Uma lista de vértices da EC

ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]

https://doi.org/10.1016%2F0020-0190%2872%2990045-2

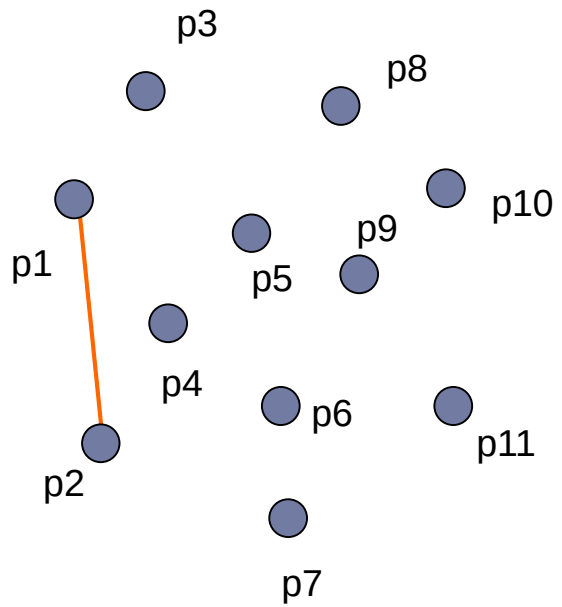
- Classificar os pontos em x, obtendo uma sequencia ordenada $p_1, p_2, \ldots, p_n$
- Calcular a EC superior percorrendo a lista acima da esquerda para a direita
- Calcular a EC inferior percorrendo a lista acima da direita para a esquerda



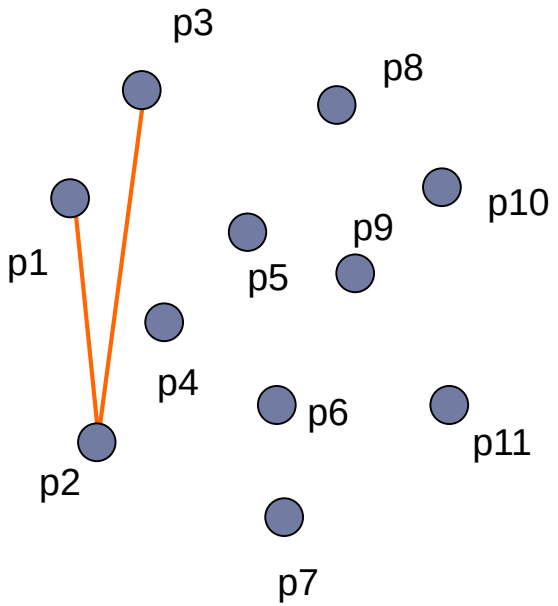
ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]

- Definindo curva [virada] para esquerda ou direita
  - Dado três pontos (x1,y1), (x2,y2) e (x3,y3), simplesmente calculando o produto vetorial (x2 − x1)(y3 − y1) − (y2 − y1) (x3 − x1) dos dois vetores definidos pelos pontos (x1,y1), (x2,y2) e (x2,y2), (x3,y3). Se o resultado for Zero, os três pontos são colineares, se for positivo, os três pontos constituem uma "curva para esquerda", caso contrario uma "curva para direita"

ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]

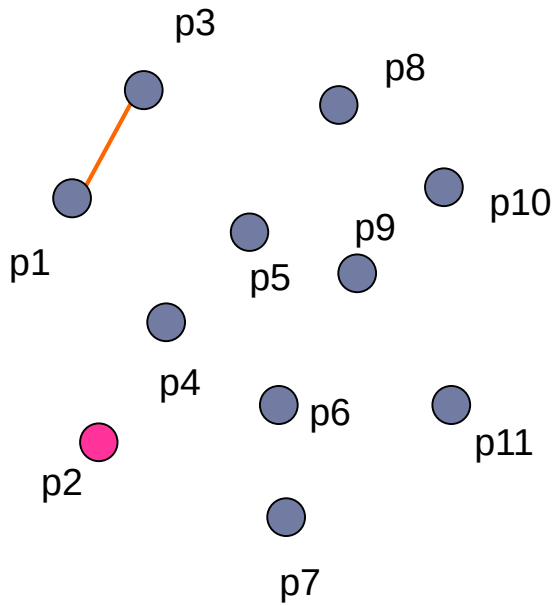


ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]

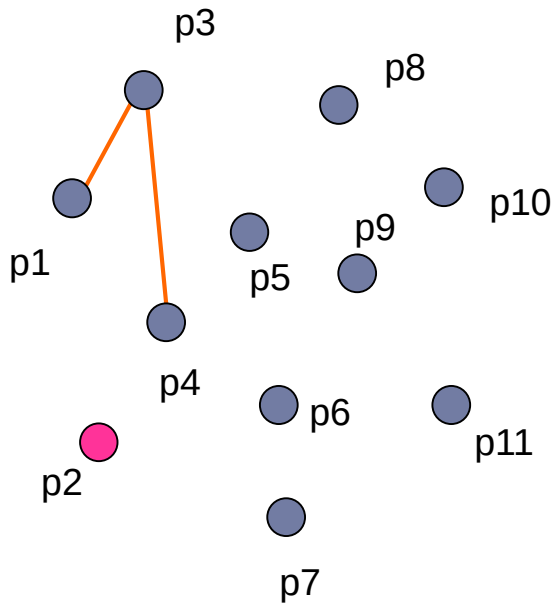


Virada p/ esquerda

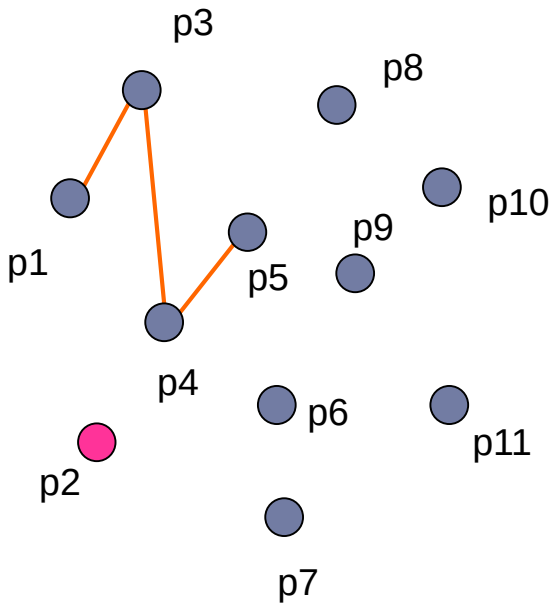
ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]



ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]

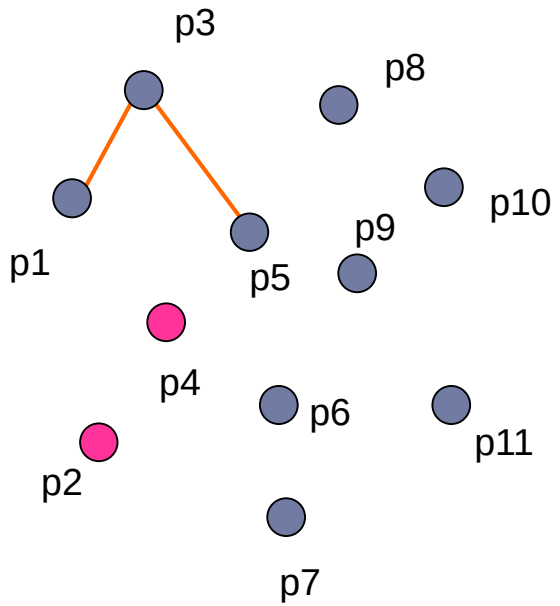


ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]

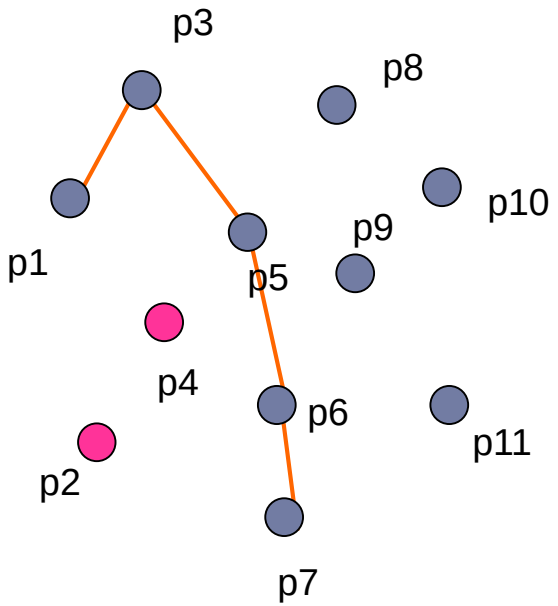


Virada p/ esquerda

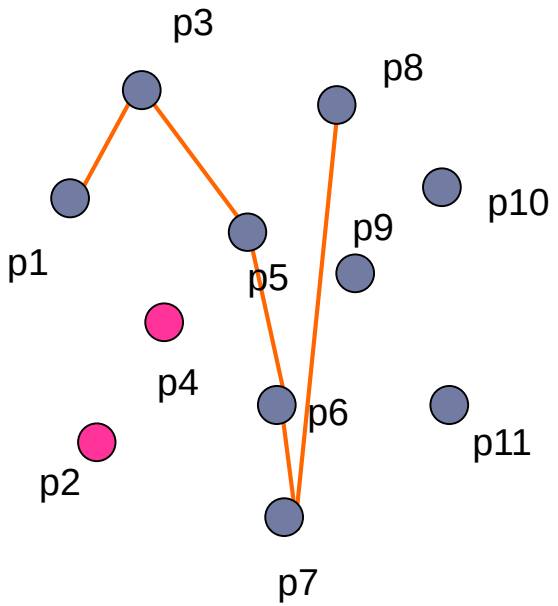
ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]



ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]

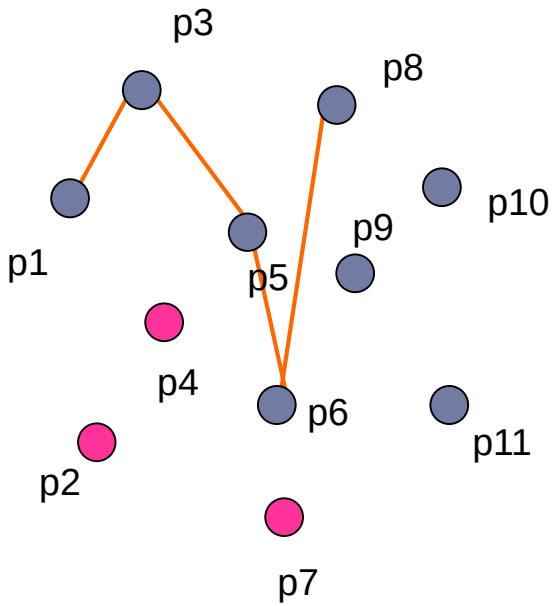


ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]



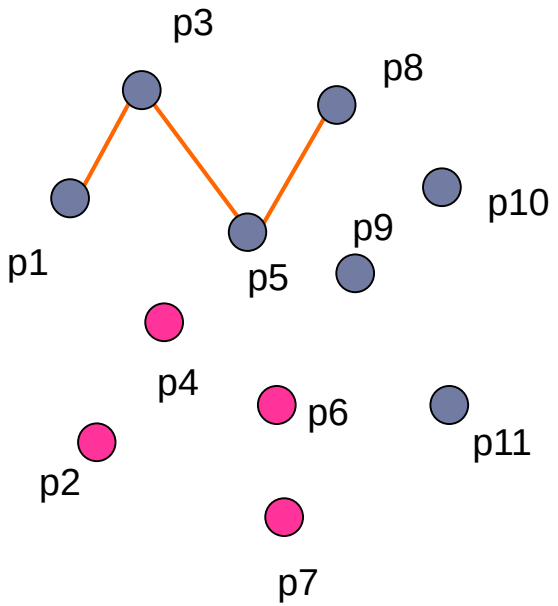
Virada p/ esquerda

ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]



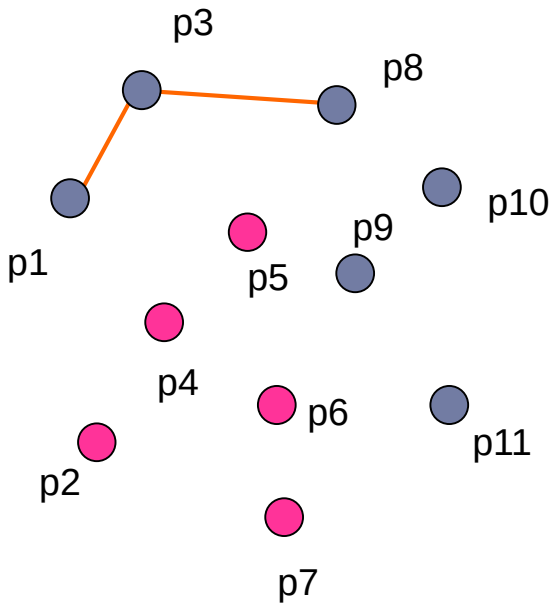
Virada p/ esquerda

ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]



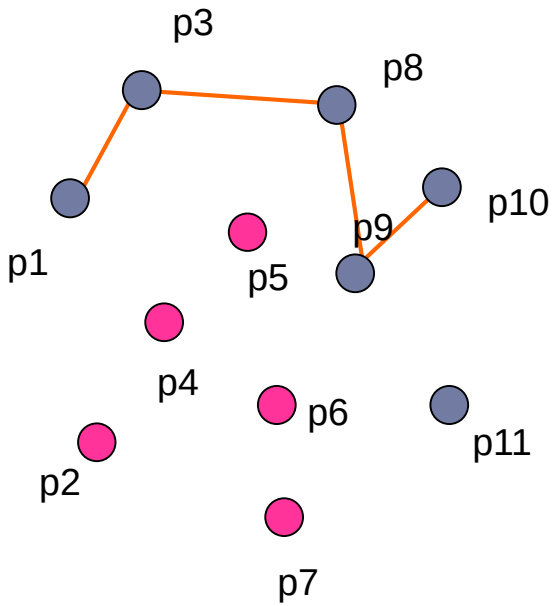
Virada p/ esquerda

ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]



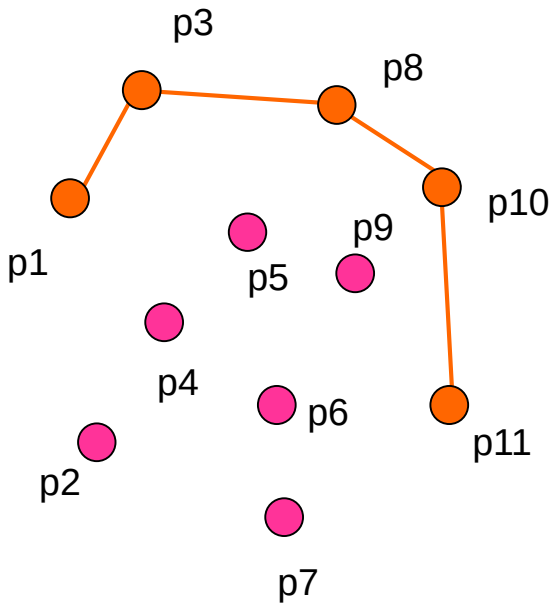
Virada p/ esquerda

ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]



Virada p/ esquerda

ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]



ENVOLTÓRIA CONVEXA
[GRAHAN'S SCAN 1972]

- **Entrada:** Conjunto de pontos P no plano

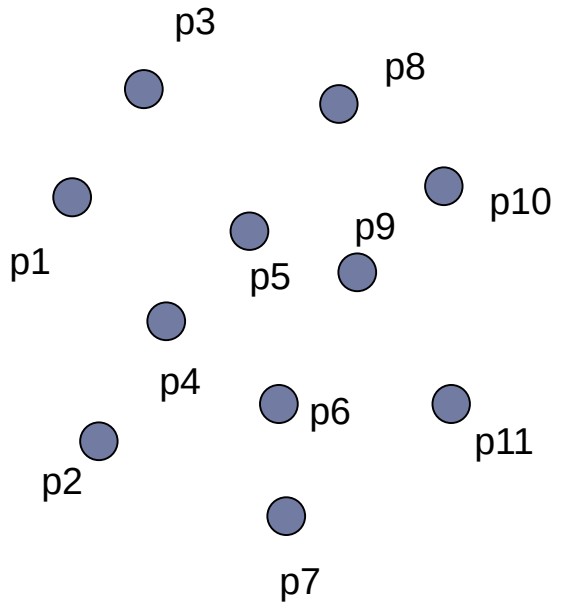
Algoritmo Grahan's Scan (EC2)

```
01.   Classificar os pontos pela coordenada X (p1, p2,... Pn)
02.   Colocar p1 e p2 na lista L_SUP
03.   FOR i:= 3 TO N
04.        DO adicionar pi a L_SUP
05.             WHILE L_SUP > 2 pontos e últimos três fazem uma
                  virada para esquerda
06.             DO delete o penultimo ponto de L_SUP
07.   Coloque os pontos pn e pn-1 na lista L_INF
08.   FOR i: n-2 DOWNTO 1
09.             Repita passoas 4-6 com L_INF no lugar de L_SUP
10.   Remover o primeiro e ultimo pontos de L_SUP e L_INF
11.   Devolve concatenacao de L_SUP e L_INF
```

ENVOLTÓRIA CONVEXA
[GIFT WRAPPING – JARVIS, 1973]

https://doi.org/10.1016%2F0020-0190%2873%2990020-3

1. Find point with smallest y coordinate
2. Find edge that minimizes counter-clockwise angle with respect to supporting line
3. Repeat until reach last vertex

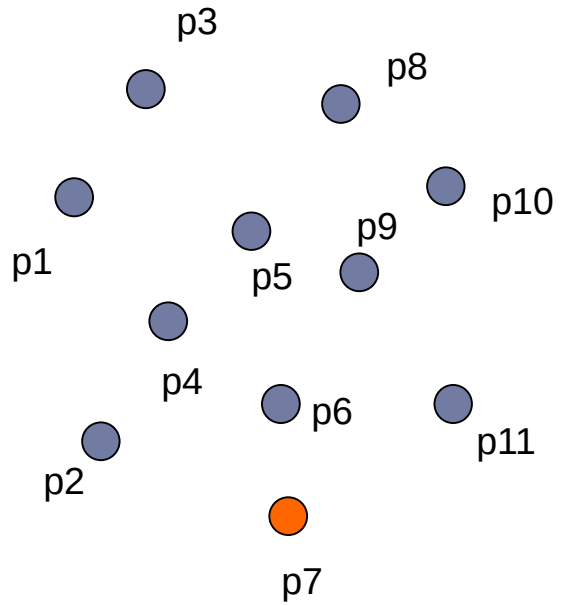


ENVOLTÓRIA CONVEXA
[GIFT WRAPPING]

1. Find point with smallest y coordinate

2. Find edge that minimizes counter-clockwise angle with respect to supporting line

3. Repeat until reach last vertex

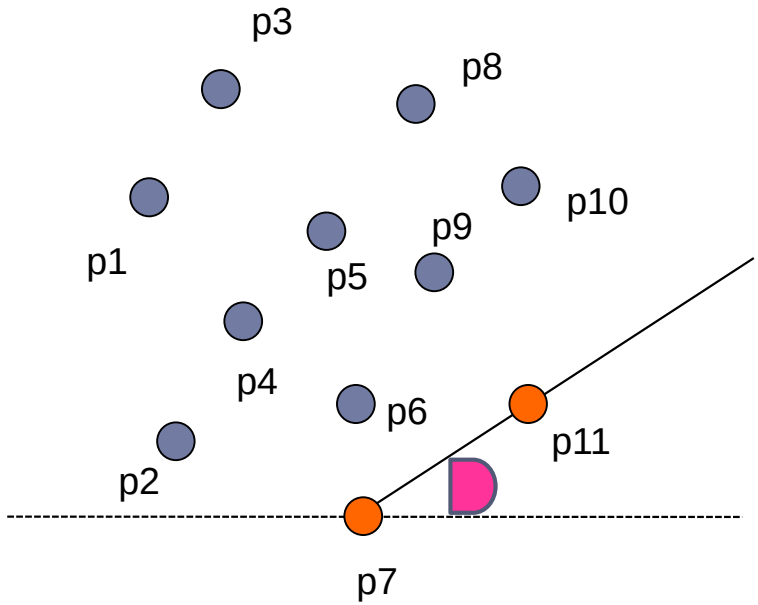


ENVOLTÓRIA CONVEXA
[GIFT WRAPPING]

1. Find point with smallest y coordinate

2. Find edge that minimizes counter-clockwise angle with respect to supporting line

3. Repeat until reach last vertex

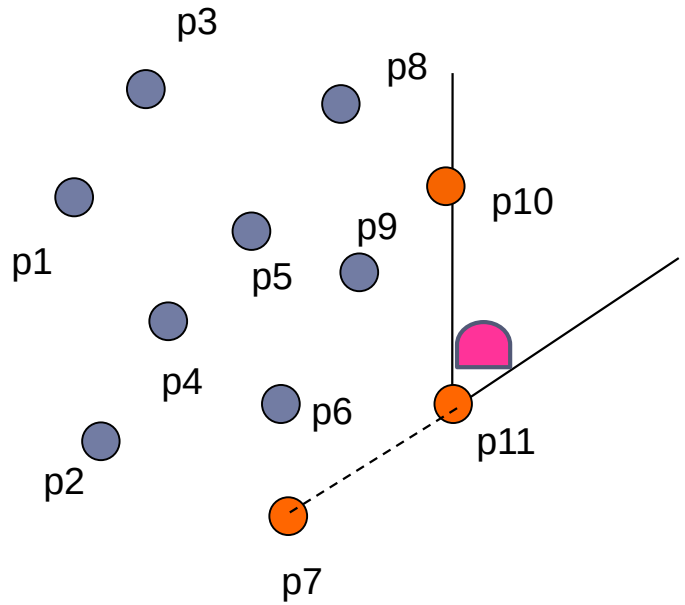


ENVOLTÓRIA CONVEXA
[GIFT WRAPPING]

1. Find point with smallest y coordinate

2. Find edge that minimizes counter-clockwise angle with respect to supporting line

3. Repeat until reach last vertex

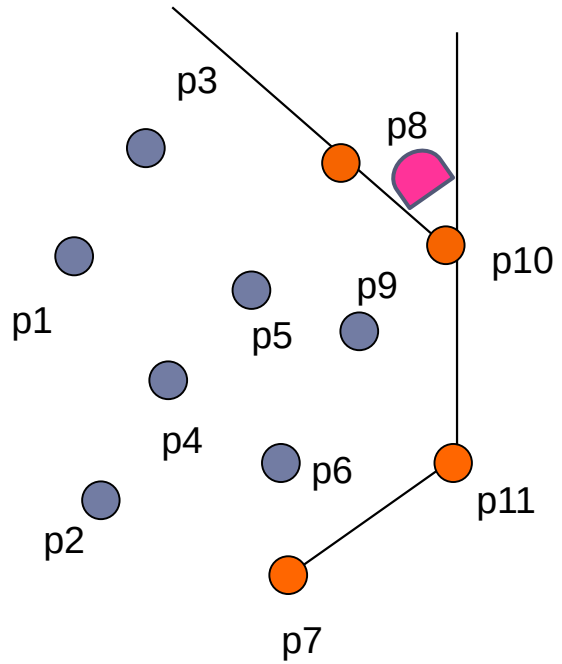


ENVOLTÓRIA CONVEXA
[GIFT WRAPPING]

1. Find point with smallest y coordinate

2. Find edge that minimizes counter-clockwise angle with respect to supporting line

3. Repeat until reach last vertex

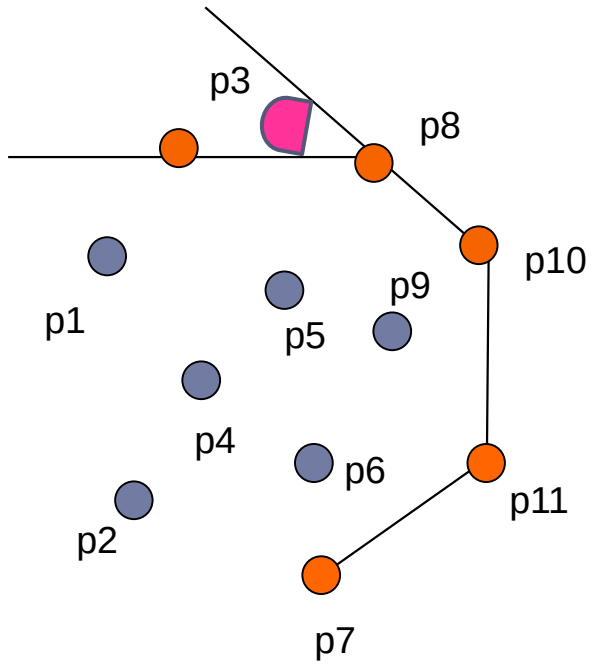


ENVOLTÓRIA CONVEXA
[GIFT WRAPPING]

1. Find point with smallest y coordinate

2. Find edge that minimizes counter-clockwise angle with respect to supporting line

3. Repeat until reach last vertex

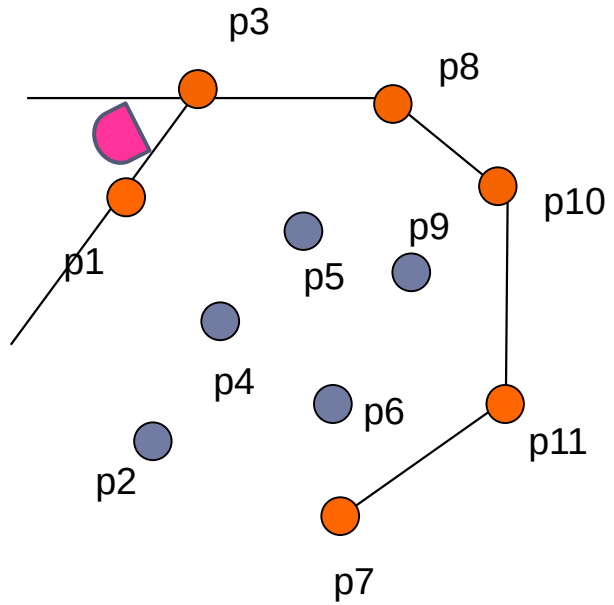


ENVOLTÓRIA CONVEXA
[GIFT WRAPPING]

1. Find point with smallest y coordinate

2. Find edge that minimizes counter-clockwise angle with respect to supporting line

3. Repeat until reach last vertex

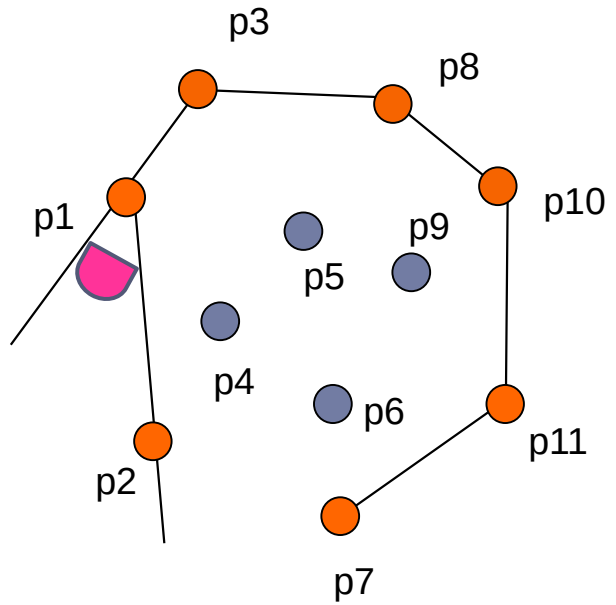


ENVOLTÓRIA CONVEXA
[GIFT WRAPPING]

1. Find point with smallest y coordinate

2. Find edge that minimizes counter-clockwise angle with respect to supporting line

3. Repeat until reach last vertex

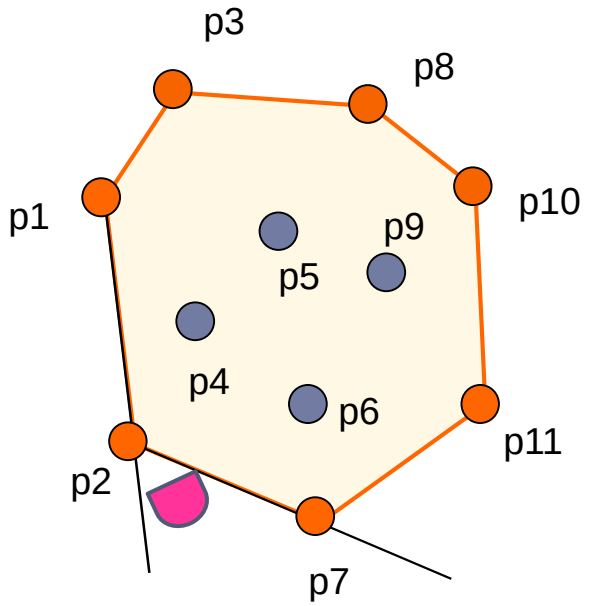


ENVOLTÓRIA CONVEXA
[GIFT WRAPPING]

1. Find point with smallest y coordinate

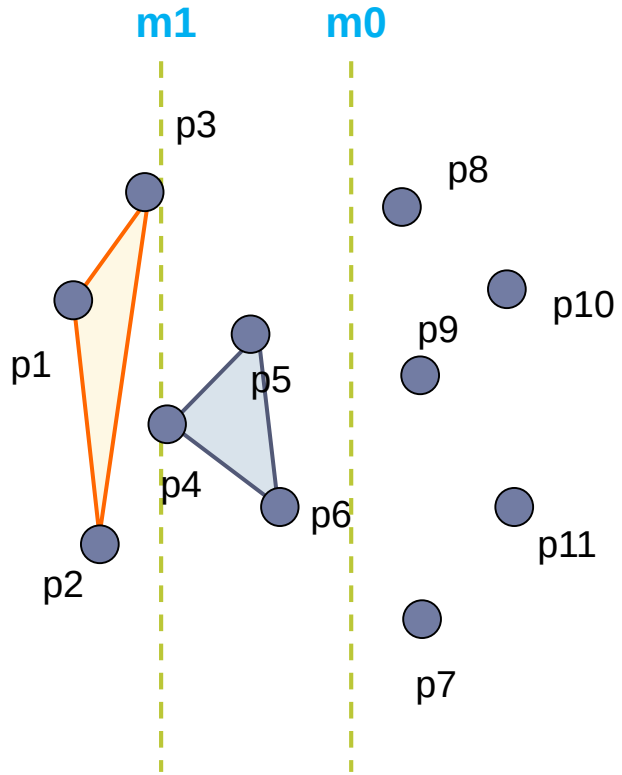
2. Find edge that minimizes counter-clockwise angle with respect to supporting line

3. Repeat until reach last vertex



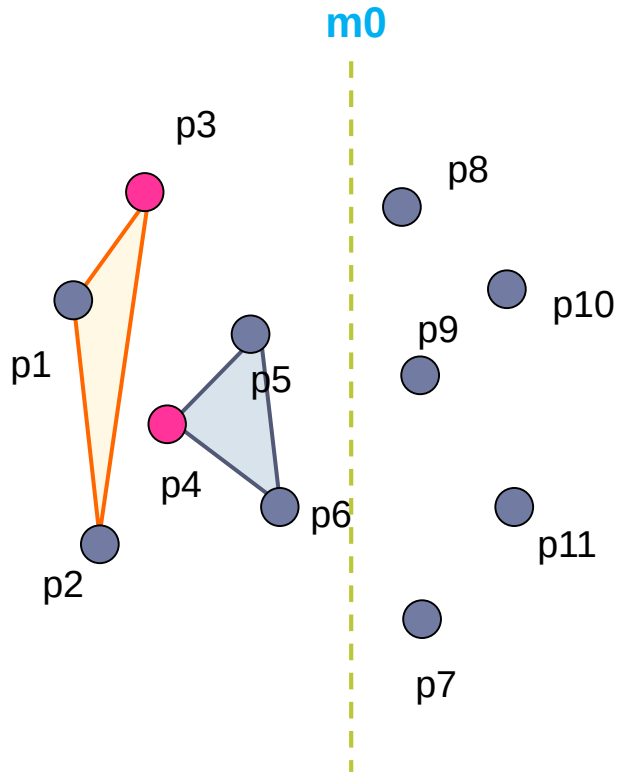
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull



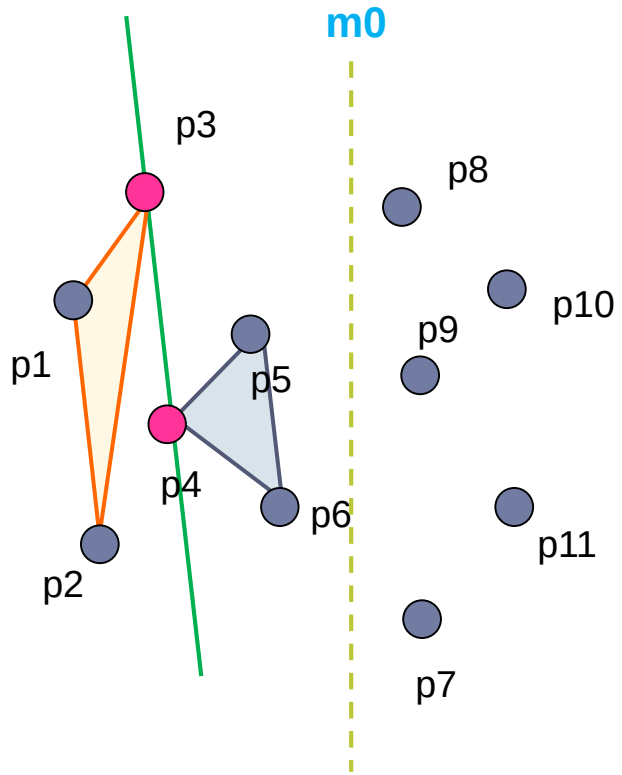
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



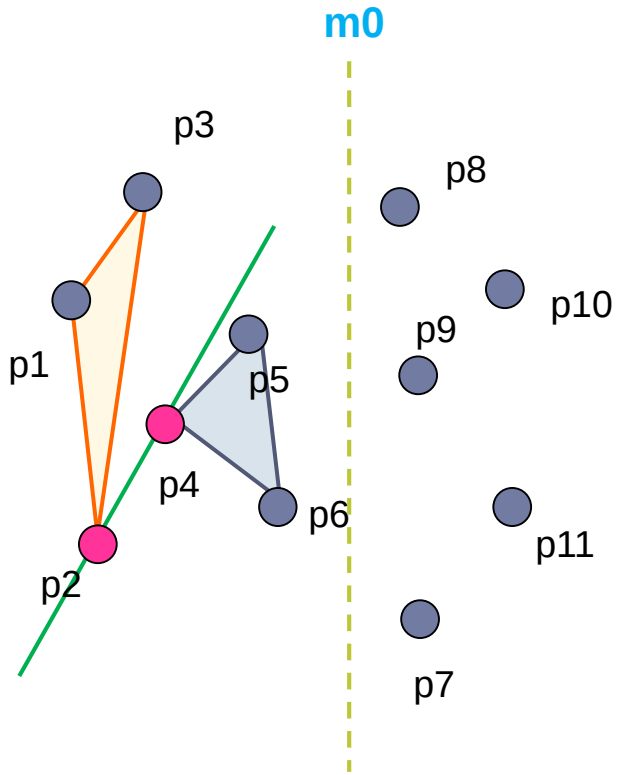
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



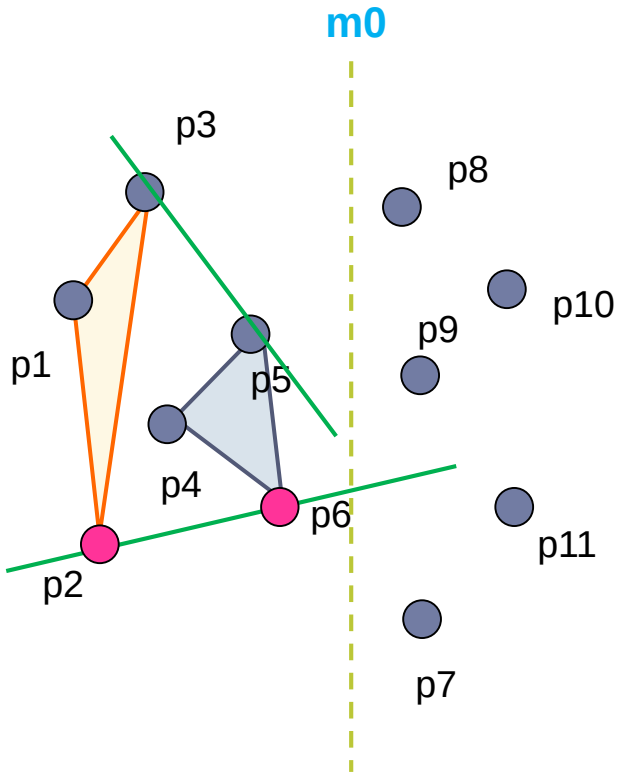
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



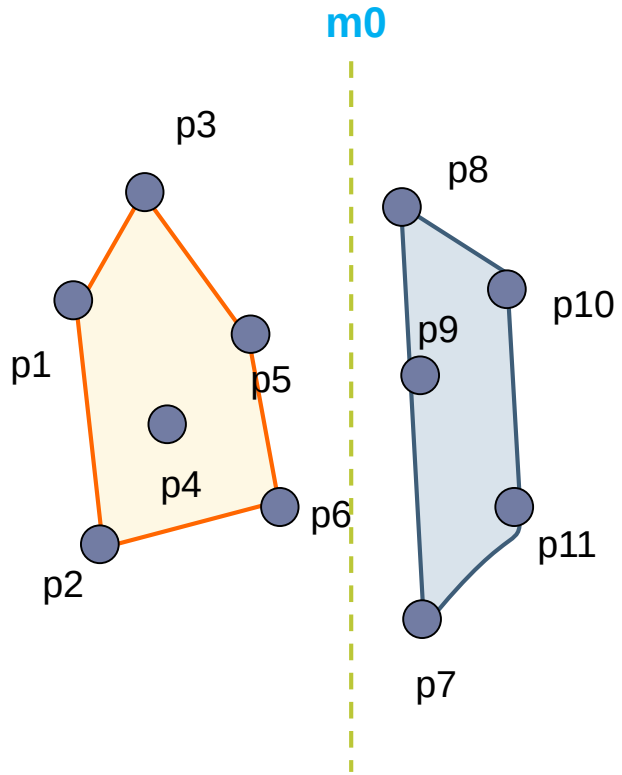
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



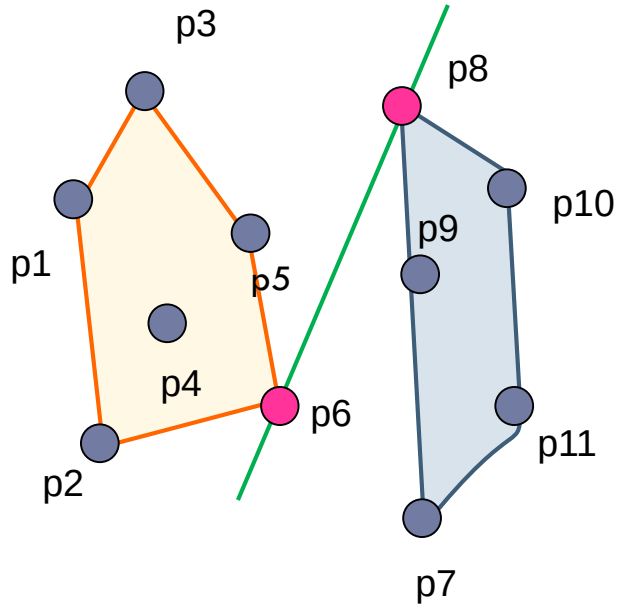
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



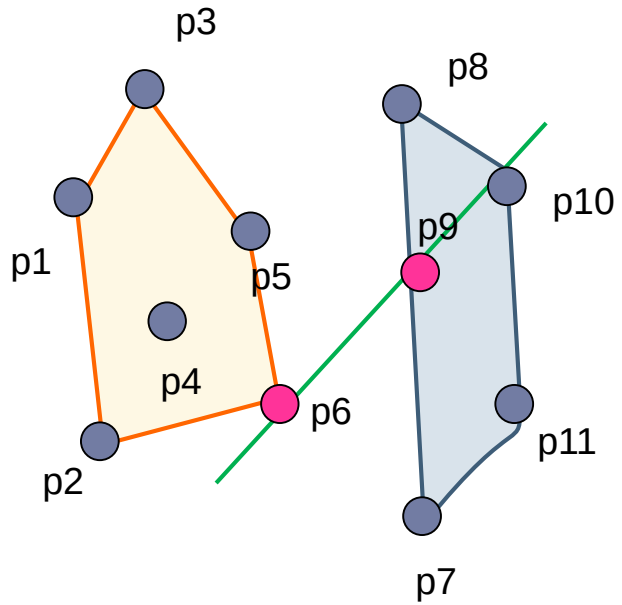
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



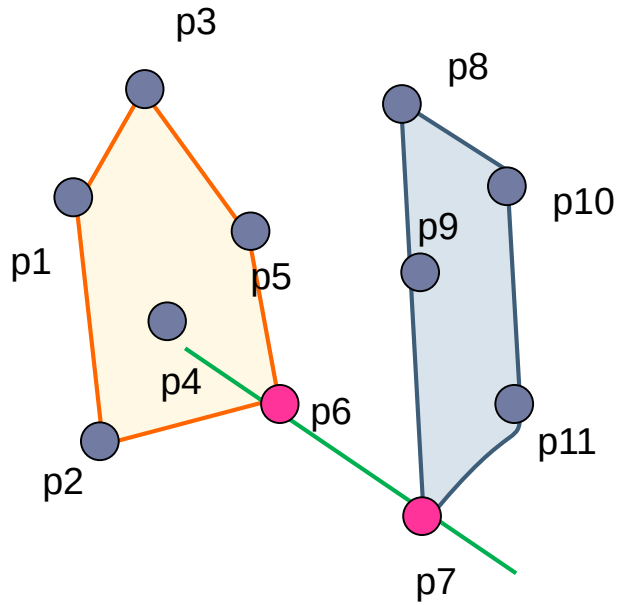
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



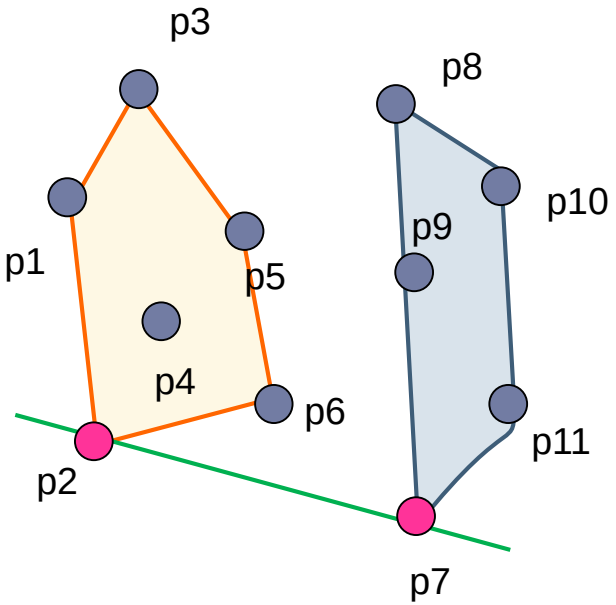
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



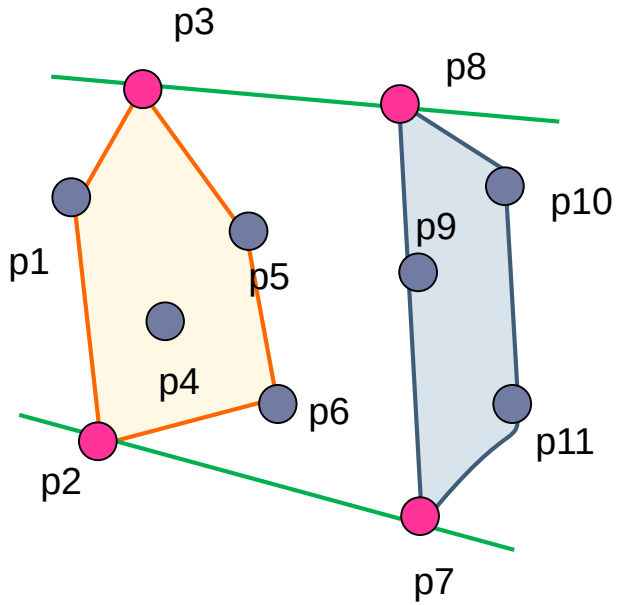
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



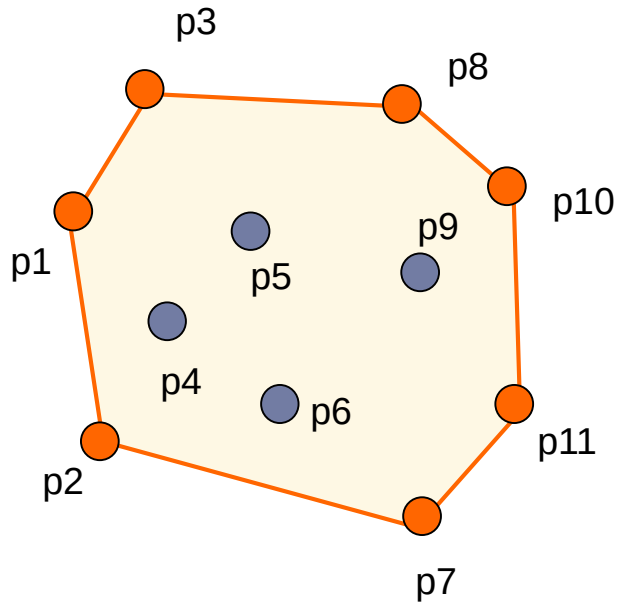
ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER – PREPARATA, 1977]

1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



ENVOLTÓRIA CONVEXA
[DIVIDE-AND-CONQUER]

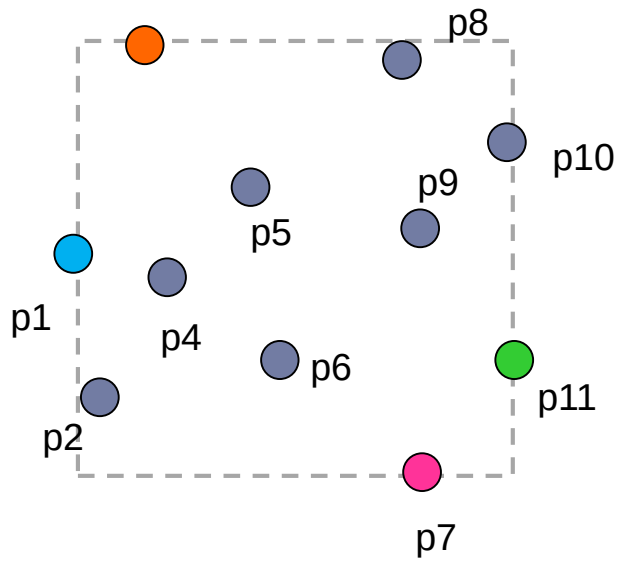
1. Sort point in x
2. Divide points in two sets A and B with similar number of elements
3. Compute Convex Hull of A and B recursively
4. Merge A and B to define convex hull
5. Find lower tangent common to A and B
6. Find rightmost point a of A and leftmost point b of B
7. While ab is not lower tangent to a, decrease a
8. While ab is not lower tangent of b, decrease b



ENVOLTÓRIA CONVEXA [QUICKHULL – BARBER, 1996]

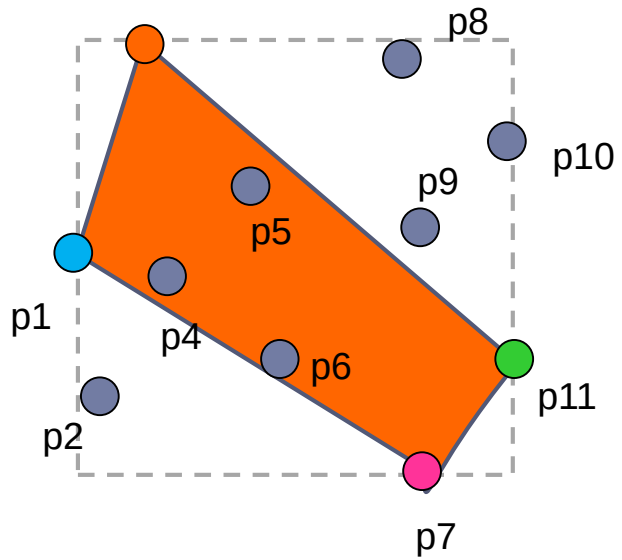
https://doi.org/10.1145%2F235815.235821

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



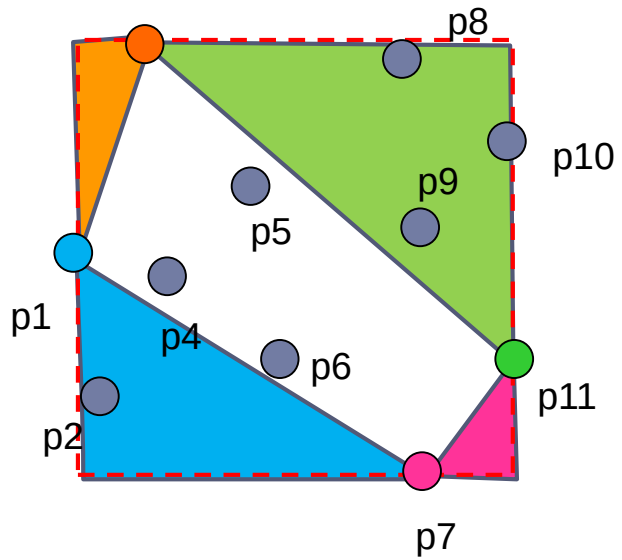
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points



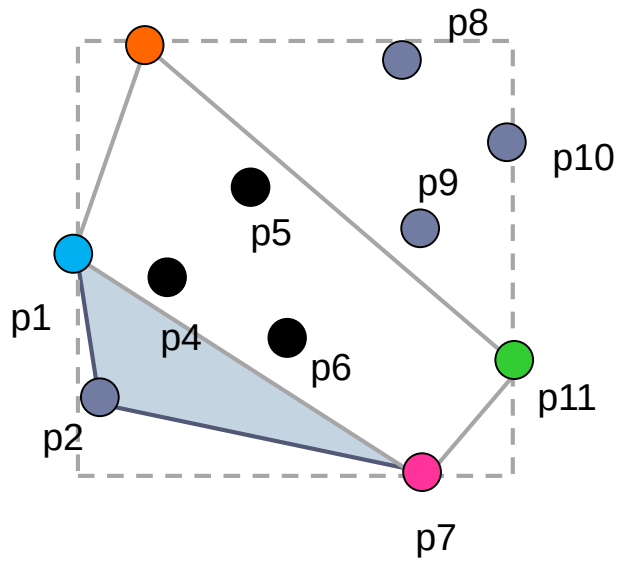
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively



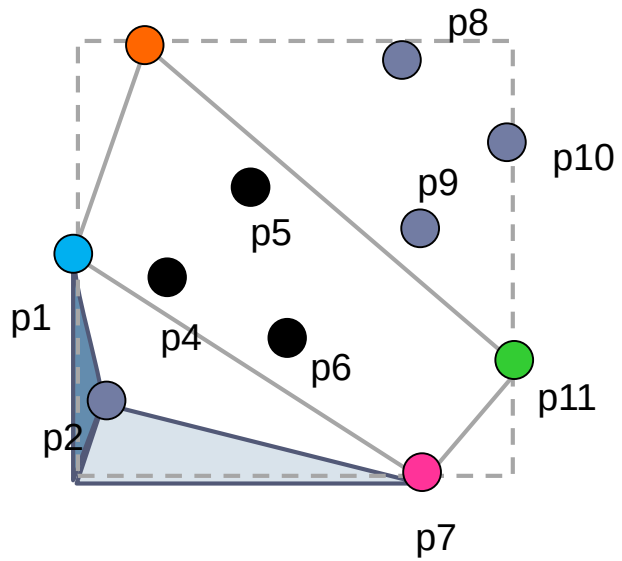
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles



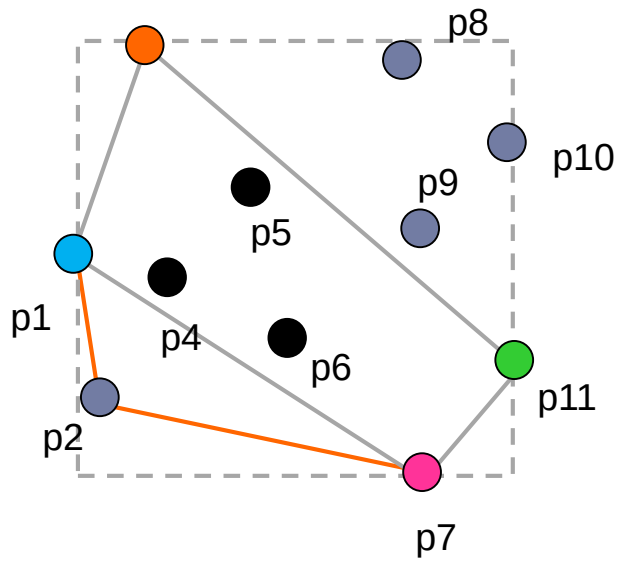
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



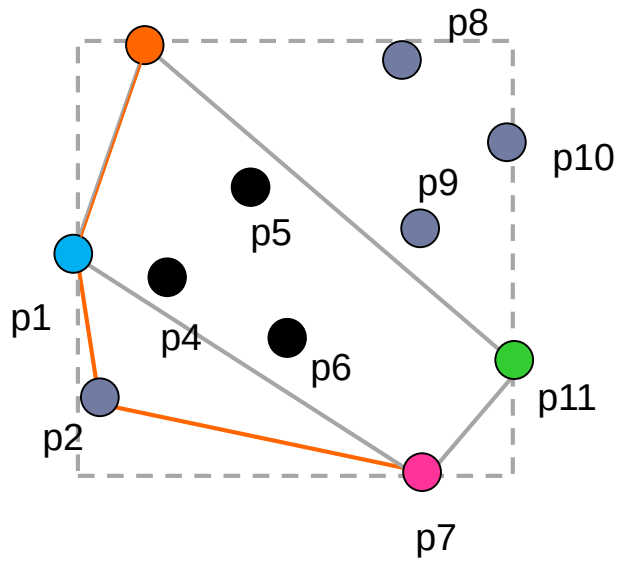
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



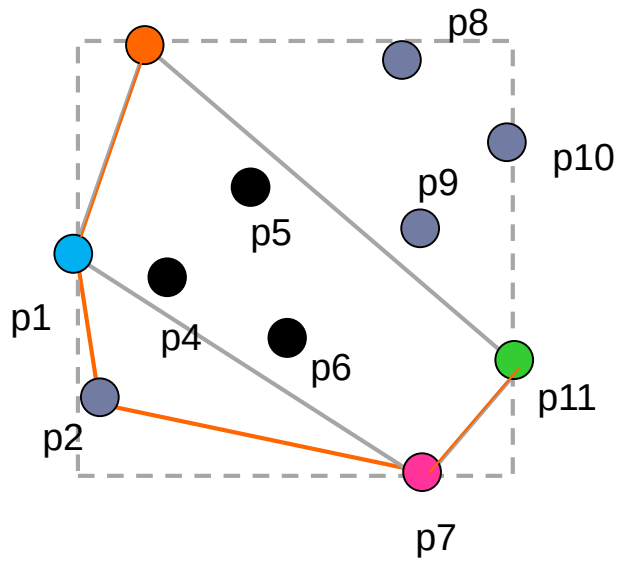
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



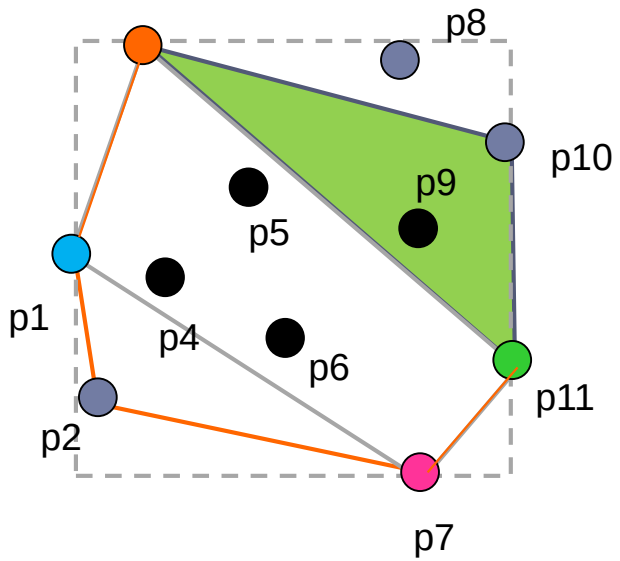
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



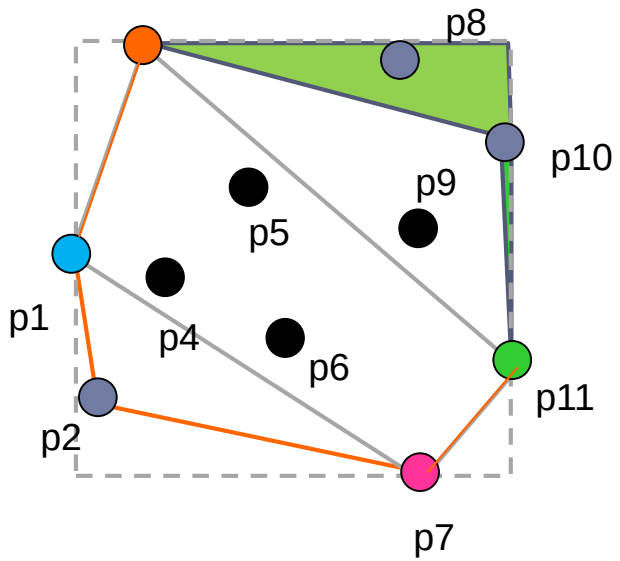
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



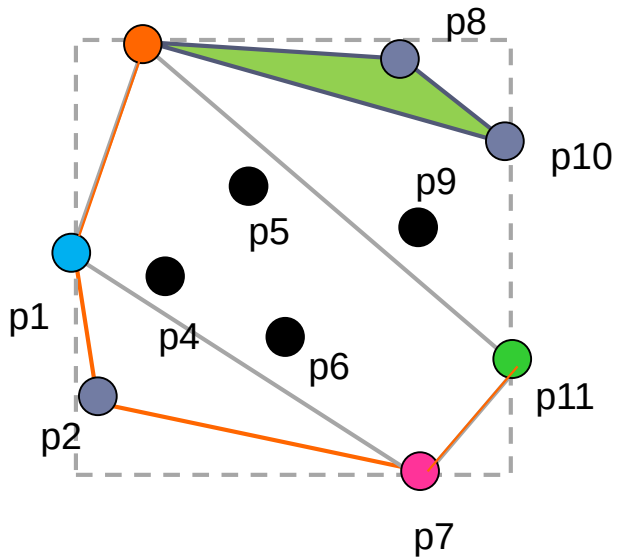
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



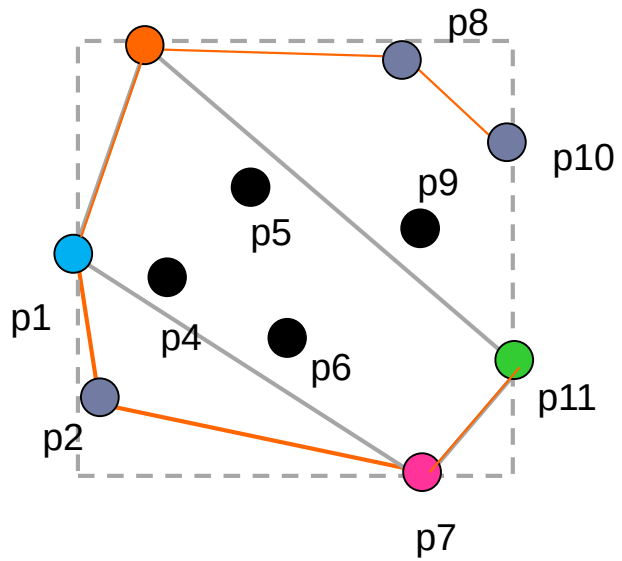
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



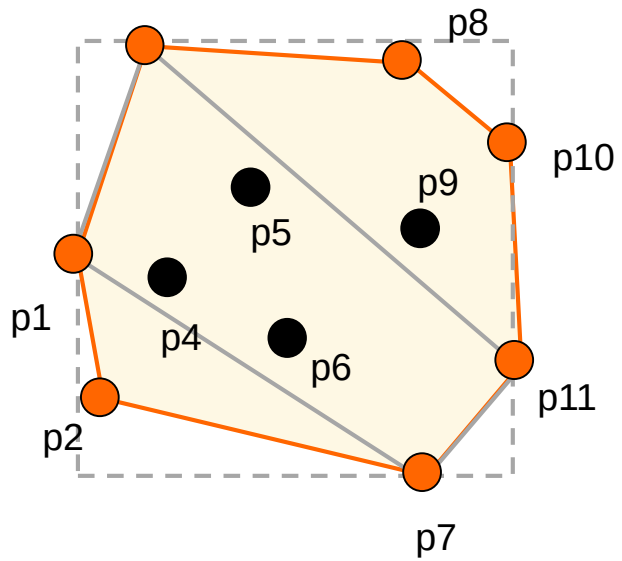
ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



ENVOLTÓRIA CONVEXA
[QUICKHULL]

1. Find extreme points above, below, left and right
2. Define a quadrilateral connecting extreme points, discard internal points
3. Process 4 triangles recursively
4. For each triangle, find vertex with greater distance from the triangle baseline
5. Define another triangle and discard internal triangles
6. Process 2 triangles recursively



Exercício

10078 - The Art Gallery

https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=1019

Cálculo do produto vetorial.

```
boolean CCW(Point p1, Point p2, Point p3){
            double value = (p2.getX()-p1.getX())*(p3.getY()-p1.getY())-
                           (p2.getY()-p1.getY())*(p3.getX()-p1.getX());
            if ( value < 0.000001 )
                    return true;
            else
                    return false;
}
```