Aula 18

Fecho convexo: quickhull

Sec 7.4.6 do FP (JAI 2009).

Combinação convexa

P: coleção de pontos do plano, dada por $X[1\,..\,n]$, $Y[1\,..\,n]$.

Combinação convexa

P: coleção de pontos do plano, dada por $X[1\,.\,.\,n]$, $Y[1\,.\,.\,n]$.

Combinação convexa de pontos de P: soma da forma

$$\alpha_1(X[1], Y[1]) + \cdots + \alpha_n(X[n], Y[n]),$$

com $\alpha_i \geq 0$, para $i = 1, \ldots, n$, e $\alpha_1 + \cdots + \alpha_n = 1$.

Combinação convexa

P: coleção de pontos do plano, dada por $X[1\,.\,.\,n]$, $Y[1\,.\,.\,n]$.

Combinação convexa de pontos de P: soma da forma

$$\alpha_1(X[1], Y[1]) + \cdots + \alpha_n(X[n], Y[n]),$$

com $\alpha_i \geq 0$, para $i = 1, \ldots, n$, e $\alpha_1 + \cdots + \alpha_n = 1$.

Fecho convexo de P:
conjunto de combinações convexas de pontos de P, ou seja,

$$\mathrm{conv}(P) := \{\alpha_1(X[1], Y[1]) + \cdots + \alpha_n(X[n], Y[n]) : \\ \alpha_1 + \cdots + \alpha_n = 1, \text{ e } \alpha_i \geq 0 \ (i = 1, \ldots, n)\}.$$

Combinação convexa

P: coleção de pontos do plano, dada por $X[1 \,..\, n], Y[1 \,..\, n]$.

Combinação convexa de pontos de P: soma da forma

$$\alpha_1(X[1], Y[1]) + \cdots + \alpha_n(X[n], Y[n]),$$

com $\alpha_i \geq 0$, para $i = 1, \ldots, n$, e $\alpha_1 + \cdots + \alpha_n = 1$.

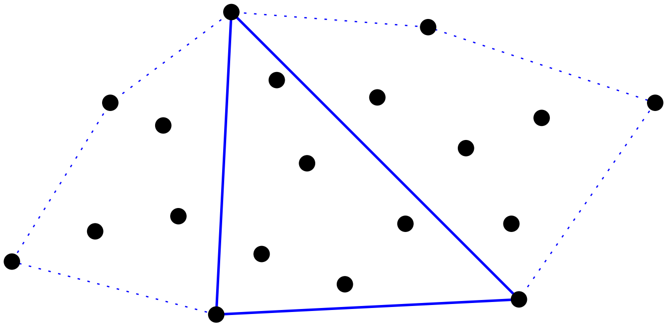
Fecho convexo de P:
conjunto de combinações convexas de pontos de P, ou seja,

$$\mathrm{conv}(P) := \{\alpha_1(X[1], Y[1]) + \cdots + \alpha_n(X[n], Y[n]) : \\ \alpha_1 + \cdots + \alpha_n = 1, \text{ e } \alpha_i \geq 0 \ (i = 1, \ldots, n)\}.$$

Problema: Dada uma coleção P de pontos do plano, determinar o fecho convexo de P.

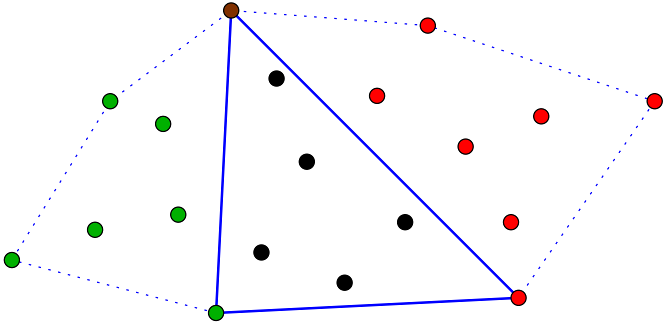
Quickhull

Ideia: descartar pontos que estão no interior do fecho convexo e concentrar o trabalho nos pontos que estão próximos da fronteira.



Quickhull

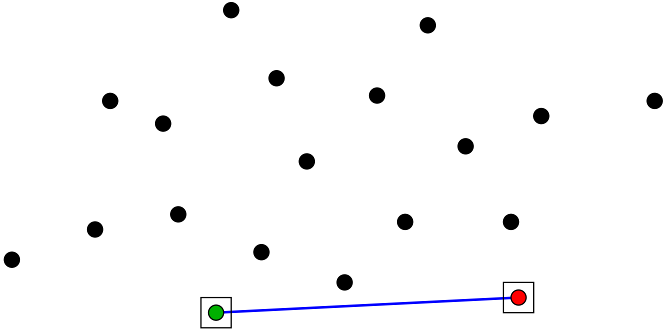
Ideia: descartar pontos que estão no interior do fecho convexo e concentrar o trabalho nos pontos que estão próximos da fronteira.



Monta coleções da esquerda e da direita
e aplica recursão nelas.

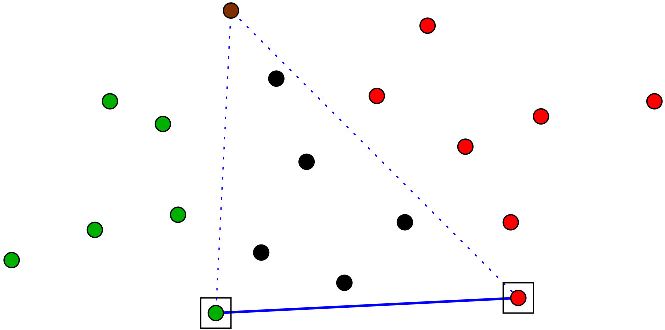
Como dividir a coleção?

Começamos com dois extremos consecutivos do fecho.



Como dividir a coleção?

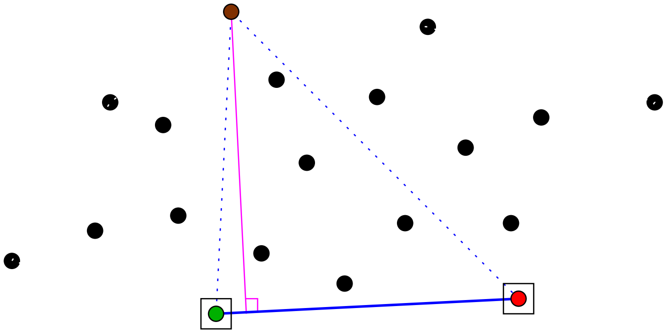
Começamos com dois extremos consecutivos do fecho.



Então precisamos encontrar o extremo marrom
e dividir em três a coleção de pontos.

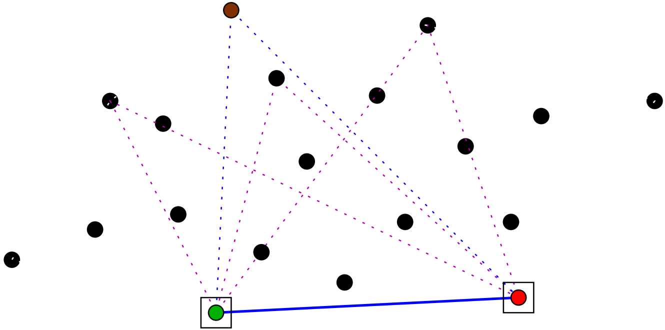
Como encontrar o ponto marrom?

O extremo marrom é o ponto mais distante da coleção em relação à reta que passa pelos dois extremos iniciais.



Como encontrar o ponto marrom?

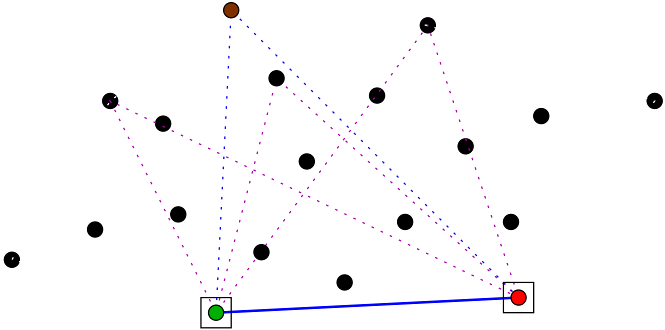
O extremo marrom é o ponto mais distante da coleção em relação à reta que passa pelos dois extremos iniciais.



Como são as áreas dos vários triângulos?
Qual tem a maior área?

Como encontrar o ponto marrom?

O extremo marrom é o ponto mais distante da coleção em relação à reta que passa pelos dois extremos iniciais.



Como são as áreas dos vários triângulos?
Qual tem a maior área?
O triângulo com terceira ponta mais distante!

O ponto marrom

$X[p\,..\,r], Y[p\,..\,r]$: coleção com ≥ 3 pontos em posição geral.

O ponto marrom

$X[p\,.\,.\,r], Y[p\,.\,.\,r]$: coleção com ≥ 3 pontos em posição geral.

A função abaixo devolve a área do triângulo
cujos extremos são os pontos de índices i, j e k.

Área (X, Y, i, j, k)
1 devolva $|\mathrm{Det}(X[i], Y[i], X[j], Y[j], X[k], Y[k])|/2$

O ponto marrom

$X[p\,..\,r], Y[p\,..\,r]$: coleção com ≥ 3 pontos em posição geral.

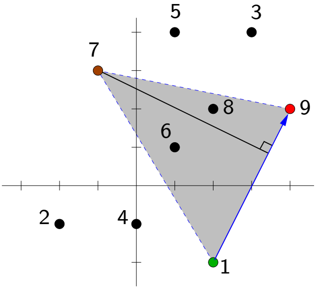
A função abaixo devolve a área do triângulo
cujos extremos são os pontos de índices i, j e k.

Área (X, Y, i, j, k)
devolva $|\mathrm{Det}(X[i], Y[i], X[j], Y[j], X[k], Y[k])|/2$

Recebe $X[p\,..\,r], Y[p\,..\,r]$ e, usando Área, devolve
o índice de um ponto extremo da coleção distinto de p e r.

PontoExtremo (X, Y, p, r)
$q \leftarrow p + 1$ $\quad$ *maior* $\leftarrow$ Área(X, Y, p, r, q)
para $i \leftarrow p + 2$ até $r - 1$ faça
$\quad$ se Área$(X, Y, p, r, i) >$ *maior*
$\quad\quad$ então $q \leftarrow i$ $\quad$ *maior* $\leftarrow$ Área(X, Y, p, r, q)
devolva q

Exemplo

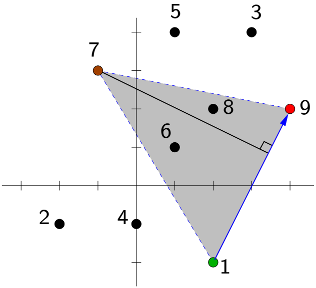


X	2	−2	3	0	1	1	−1	2	4
	1	2	3	4	5	6	7	8	9

Y	−2	−1	4	−1	4	1	3	2	2
	1	2	3	4	5	6	7	8	9

$\text{PontoExtremo}(X, Y, 1, 9) = 7$

Exemplo



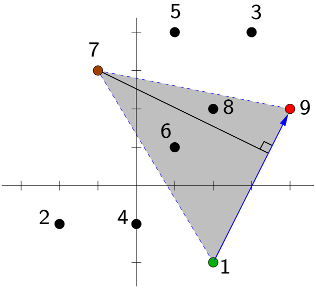
X	2	−2	3	0	1	1	−1	2	4
	1	2	3	4	5	6	7	8	9

Y	−2	−1	4	−1	4	1	3	2	2
	1	2	3	4	5	6	7	8	9

$\text{PontoExtremo}(X, Y, 1, 9) = 7$

Exercício: Mostre que o algoritmo de fato devolve o índice de um ponto extremo da coleção $X[p\,..\,r]$, $Y[p\,..\,r]$.

Exemplo



X	2	−2	3	0	1	1	−1	2	4
	1	2	3	4	5	6	7	8	9

Y	−2	−1	4	−1	4	1	3	2	2
	1	2	3	4	5	6	7	8	9

PontoExtremo$(X, Y, 1, 9) = 7$

Exercício: Mostre que o algoritmo de fato devolve o índice de um ponto extremo da coleção $X[p \,.\,.\, r], Y[p \,.\,.\, r]$.

Consumo de tempo de PontoExtremo (X, Y, p, r):
$\Theta(n)$, onde $n := r - p + 1$.

Particionamento

Particione (X, Y, p, r):

Recebe coleção $X[p \mathinner{..} r]$, $Y[p \mathinner{..} r]$ de pontos em posição geral, com pelo menos três pontos, tal que os pontos de índice p e r são extremos consecutivos na fronteira do fecho convexo da coleção no sentido anti-horário.

Particionamento

Particione (X, Y, p, r):

Recebe coleção $X[p\,..\,r]$, $Y[p\,..\,r]$ de pontos em posição geral, com pelo menos três pontos, tal que os pontos de índice p e r são extremos consecutivos na fronteira do fecho convexo da coleção no sentido anti-horário.

Rearranja $X[p\,..\,r]$, $Y[p\,..\,r]$ e devolve p', q tq $p \leq p' < q < r$ e

(i) o ponto de índice r permanece na mesma posição, enquanto que o ponto de índice p foi para a posição p',

(ii) o ponto de índice q é extremo,

(iii) $X[p\,..\,p'-1]$, $Y[p\,..\,p'-1]$ é uma coleção de pontos interiores ao fecho convexo de $X[p\,..\,r]$, $Y[p\,..\,r]$,

(iv) $\cdots$

Particionamento

Particione (X, Y, p, r):

Rearranja $X[p \,.\,.\, r], Y[p \,.\,.\, r]$ e devolve p', q tq $p \leq p' < q < r$ e

(i) o ponto de índice r permanece na mesma posição, enquanto que o ponto de índice p foi para a posição p',

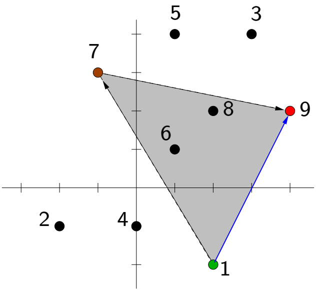
(ii) o ponto de índice q é extremo,

(iii) $X[p \,.\,.\, p'-1], Y[p \,.\,.\, p'-1]$ é uma coleção de pontos interiores ao fecho convexo de $X[p \,.\,.\, r], Y[p \,.\,.\, r]$,

(iv) $X[p'+1 \,.\,.\, q-1], Y[p'+1 \,.\,.\, q-1]$ é a coleção dos pontos que estão à esquerda da reta orientada determinada por $(X[p'], Y[p'])$ e $(X[q], Y[q])$,

(v) $X[q+1 \,.\,.\, r-1], Y[q+1 \,.\,.\, r-1]$ é a coleção dos pontos que estão à esquerda da reta orientada determinada por $(X[q], Y[q])$ e $(X[r], Y[r])$.

Particionamento

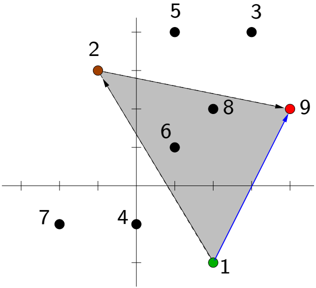


Particione(X, Y, 1, 9)

	p						q		r
X	2	−2	3	0	1	1	−1	2	4
Y	−2	−1	4	−1	4	1	3	2	2
	1	2	3	4	5	6	7	8	9

Encontra o ponto extremo indicado por q.

Particionamento



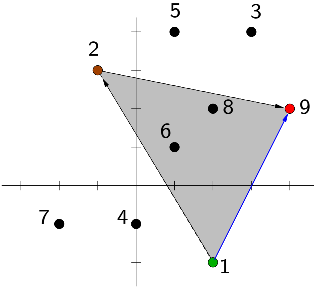
Particione(X, Y, 1, 9)

	p	$p+1$						k	p' q r
X	2	-1	3	0	1	1	-2	2	4
Y	-2	3	4	-1	4	1	-1	2	2
	1	2	3	4	5	6	7	8	9

Encontra o ponto extremo indicado por q.

Coloca esse ponto na posição $p+1$.

Particionamento



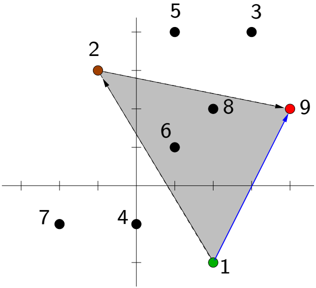
Particione$(X, Y, 1, 9)$

	p	$p{+}1$						k	p' q r
X	2	-1	3	0	1	1	-2	2	4
Y	-2	3	4	-1	4	1	-1	2	2
	1	2	3	4	5	6	7	8	9

Encontra o ponto extremo indicado por q.
Coloca esse ponto na posição $p{+}1$.

Invariante:
$X[q\,..\,r], Y[q\,..\,r]$: pontos vermelhos examinados
$X[p'\,..\,q{-}1], Y[p'\,..\,q{-}1]$: pontos verdes examinados
$X[k{+}1\,..\,p'{-}1], Y[k{+}1\,..\,p'{-}1]$: pontos interiores examinados

Particionamento



Particione$(X, Y, 1, 9)$

	p	$p{+}1$						k	p' q r
X	2	-1	3	0	1	1	-2	2	4
Y	-2	3	4	-1	4	1	-1	2	2
	1	2	3	4	5	6	7	8	9

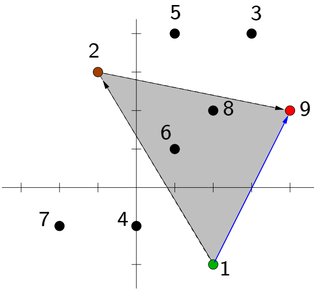
Invariante:

$X[q\,.\,.\,r], Y[q\,.\,.\,r]$: pontos vermelhos examinados

$X[p'\,.\,.\,q{-}1], Y[p'\,.\,.\,q{-}1]$: pontos verdes examinados

$X[k{+}1\,.\,.\,p'{-}1], Y[k{+}1\,.\,.\,p'{-}1]$: pontos interiores examinados

Particionamento



Particione(X, Y, 1, 9)

	p	$p+1$						k	p' q r
X	2	-1	3	0	1	1	-2	2	4
Y	-2	3	4	-1	4	1	-1	2	2
	1	2	3	4	5	6	7	8	9

Invariante:

$X[q\,..\,r]$, $Y[q\,..\,r]$: pontos vermelhos examinados

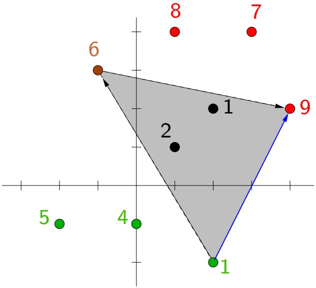
$X[p'\,..\,q-1]$, $Y[p'\,..\,q-1]$: pontos verdes examinados

$X[k+1\,..\,p'-1]$, $Y[k+1\,..\,p'-1]$: pontos interiores examinados

Para $k \leftarrow r-1$ até $p+1$

coloque o k-ésimo ponto na parte apropriada.

Particionamento



	p	$p{+}1$						k	p' q r
X	2	−1	3	0	1	1	−2	2	4
Y	−2	3	4	−1	4	1	−1	2	2

	p		p'			q			r
X	2	1	2	0	−2	−1	3	1	4
Y	2	1	−2	−1	−1	3	4	4	2
	1	2	3	4	5	6	7	8	9

Ao final...

Invariante:

$X[q\,..\,r], Y[q\,..\,r]$: pontos vermelhos examinados

$X[p'\,..\,q{-}1], Y[p'\,..\,q{-}1]$: pontos verdes examinados

$X[k{+}1\,..\,p'{-}1], Y[k{+}1\,..\,p'{-}1]$: pontos interiores examinados

Particione

Particione (X, Y, p, r)

$q \leftarrow$ PontoExtremo(X, Y, p, r)

$(X[p{+}1], Y[p{+}1]) \leftrightarrow (X[q], Y[q])$

$p' \leftarrow r \quad q \leftarrow r$

para $k \leftarrow r - 1$ decrescendo até $p + 2$ faça

Particione

Particione (X, Y, p, r)

```
q ← PontoExtremo(X, Y, p, r)
(X[p+1], Y[p+1]) ↔ (X[q], Y[q])
p' ← r     q ← r
para k ← r − 1 decrescendo até p + 2 faça
    se Esq(X, Y, p, p+1, k)        ▷ verde?
```

Particione

Particione (X, Y, p, r)

$q \leftarrow$ PontoExtremo(X, Y, p, r)
$(X[p{+}1], Y[p{+}1]) \leftrightarrow (X[q], Y[q])$
$p' \leftarrow r \quad q \leftarrow r$
para $k \leftarrow r - 1$ decrescendo até $p + 2$ faça
    se Esq$(X, Y, p, p{+}1, k)$ $\quad \triangleright$ verde?
        então $p' \leftarrow p' - 1 \quad (X[p'], Y[p']) \leftrightarrow (X[k], Y[k])$

Particione

Particione (X, Y, p, r)

$q \leftarrow$ PontoExtremo(X, Y, p, r)
$(X[p{+}1], Y[p{+}1]) \leftrightarrow (X[q], Y[q])$
$p' \leftarrow r \quad q \leftarrow r$
para $k \leftarrow r - 1$ decrescendo até $p + 2$ faça
    se Esq$(X, Y, p, p{+}1, k)$ $\quad \triangleright$ verde?
        então $p' \leftarrow p' - 1 \quad (X[p'], Y[p']) \leftrightarrow (X[k], Y[k])$
        senão se Esq$(X, Y, p{+}1, r, k)$ $\quad \triangleright$ vermelho?

Particione

Particione (X, Y, p, r)

$q \leftarrow$ PontoExtremo(X, Y, p, r)
$(X[p{+}1], Y[p{+}1]) \leftrightarrow (X[q], Y[q])$
$p' \leftarrow r \quad q \leftarrow r$
para $k \leftarrow r - 1$ decrescendo até $p + 2$ faça
    se Esq$(X, Y, p, p{+}1, k)$ $\quad \triangleright$ verde?
        então $p' \leftarrow p' - 1 \quad (X[p'], Y[p']) \leftrightarrow (X[k], Y[k])$
        senão se Esq$(X, Y, p{+}1, r, k)$ $\quad \triangleright$ vermelho?
                então $p' \leftarrow p' - 1 \quad q \leftarrow q - 1$
                        $(X[q], Y[q]) \leftrightarrow (X[k], Y[k])$

Particione

```
Particione (X, Y, p, r)
q ← PontoExtremo(X, Y, p, r)
(X[p+1], Y[p+1]) ↔ (X[q], Y[q])
p′ ← r    q ← r
para k ← r − 1 decrescendo até p + 2 faça
    se Esq(X, Y, p, p+1, k)        ▷ verde?
        então p′ ← p′ − 1    (X[p′], Y[p′]) ↔ (X[k], Y[k])
        senão se Esq(X, Y, p+1, r, k)        ▷ vermelho?
                then então p′ ← p′ − 1    q ← q − 1
                     (X[q], Y[q]) ↔ (X[k], Y[k])
                     se p′ ≠ q então (X[k], Y[k]) ↔ (X[p′], Y[p′])
```

Particione

Particione (X, Y, p, r)

$q \leftarrow$ PontoExtremo(X, Y, p, r)
$(X[p{+}1], Y[p{+}1]) \leftrightarrow (X[q], Y[q])$
$p' \leftarrow r \quad q \leftarrow r$
para $k \leftarrow r - 1$ decrescendo até $p + 2$ faça
    se Esq$(X, Y, p, p{+}1, k)$ $\quad \triangleright$ verde?
        então $p' \leftarrow p' - 1 \quad (X[p'], Y[p']) \leftrightarrow (X[k], Y[k])$
        senão se Esq$(X, Y, p{+}1, r, k)$ $\quad \triangleright$ vermelho?
                então $p' \leftarrow p' - 1 \quad q \leftarrow q - 1$
                        $(X[q], Y[q]) \leftrightarrow (X[k], Y[k])$
                        se $p' \neq q$ então $(X[k], Y[k]) \leftrightarrow (X[p'], Y[p'])$
$p' \leftarrow p' - 1 \quad q \leftarrow q - 1$
$(X[q], Y[q]) \leftrightarrow (X[p{+}1], Y[p{+}1])$
se $p' \neq q$ então $(X[p'], Y[p']) \leftrightarrow (X[p{+}1], Y[p{+}1])$
$p' \leftarrow p' - 1 \quad (X[p'], Y[p']) \leftrightarrow (X[p], Y[p])$
devolva (p', q)

Particione

Particione (X, Y, p, r)

$q \leftarrow$ PontoExtremo(X, Y, p, r)
$(X[p{+}1], Y[p{+}1]) \leftrightarrow (X[q], Y[q])$
$p' \leftarrow r \quad q \leftarrow r$
para $k \leftarrow r - 1$ decrescendo até $p + 2$ faça
    se Esq$(X, Y, p, p{+}1, k)$ $\quad \triangleright$ verde?
        então $p' \leftarrow p' - 1 \quad (X[p'], Y[p']) \leftrightarrow (X[k], Y[k])$
        senão se Esq$(X, Y, p{+}1, r, k)$ $\quad \triangleright$ vermelho?
            então $p' \leftarrow p' - 1 \quad q \leftarrow q - 1$
                $(X[q], Y[q]) \leftrightarrow (X[k], Y[k])$
                se $p' \neq q$ então $(X[k], Y[k]) \leftrightarrow (X[p'], Y[p'])$
$p' \leftarrow p' - 1 \quad q \leftarrow q - 1$
$(X[q], Y[q]) \leftrightarrow (X[p{+}1], Y[p{+}1])$
se $p' \neq q$ então $(X[p'], Y[p']) \leftrightarrow (X[p{+}1], Y[p{+}1])$
$p' \leftarrow p' - 1 \quad (X[p'], Y[p']) \leftrightarrow (X[p], Y[p])$
devolva (p', q)

Consumo de tempo: $\Theta(n)$, onde $n = r - p + 1$.

Quickhull

QuickHull (X, Y, n)
se $n = 1$
    então $h \leftarrow 1 \quad H[1] \leftarrow 1$
    senão $k \leftarrow \min\{i \in [1\,..\,n] : Y[i] \leq Y[j], 1 \leq j \leq n\}$
        $(X[1], Y[1]) \leftrightarrow (X[k], Y[k])$
        $i \leftarrow 2$
        para $j \leftarrow 3$ até n faça
            se $\mathrm{Dir}(X, Y, 1, i, j)$ então $i \leftarrow j$
        $(X[n], Y[n]) \leftrightarrow (X[i], Y[i])$
        $(H, h) \leftarrow \mathrm{QuickHullRec}(X, Y, 1, n)$
devolva (H, h)

Quickhull

```
QuickHull (X, Y, n)
se n = 1
    então h ← 1    H[1] ← 1
    senão k ← min{i ∈ [1 . . n] : Y[i] ≤ Y[j], 1 ≤ j ≤ n}
          (X[1], Y[1]) ↔ (X[k], Y[k])
          i ← 2
          para j ← 3 até n faça
              se Dir(X, Y, 1, i, j) então i ← j
          (X[n], Y[n]) ↔ (X[i], Y[i])
          (H, h) ← QuickHullRec(X, Y, 1, n)
devolva (H, h)
```

Consumo de tempo: $\Theta(n) + T(n)$, onde $n = r - p + 1$ e $T(n)$ é o tempo consumido por QuickHullRec$(X, Y, 1, n)$.

Miolo recursivo do Quickhull

QuickHullRec (X, Y, p, r)

se $p = r - 1$ $\triangleright$ há exatamente dois pontos na coleção

  então $h \leftarrow 2$ $H[1] \leftarrow r$ $H[2] \leftarrow p$

  senão $(p', q) \leftarrow \text{Particione}(X, Y, p, r)$

    $(H, h) \leftarrow \text{QuickHullRec}(X, Y, q, r)$

    $(H', h') \leftarrow \text{QuickHullRec}(X, Y, p', q)$

    $\triangleright$ $H \leftarrow H \cdot H'$ removendo uma cópia do q

    para $i \leftarrow 2$ até h' faça

      $h \leftarrow h + 1$ $H[h] \leftarrow H'[i]$

devolva (H, h)

Miolo recursivo do Quickhull

QuickHullRec (X, Y, p, r)

se $p = r - 1$ $\quad \triangleright$ há exatamente dois pontos na coleção
  então $h \leftarrow 2 \quad H[1] \leftarrow r \quad H[2] \leftarrow p$
  senão $(p', q) \leftarrow$ Particione(X, Y, p, r)
    $(H, h) \leftarrow$ QuickHullRec(X, Y, q, r)
    $(H', h') \leftarrow$ QuickHullRec(X, Y, p', q)
    $\triangleright$ $H \leftarrow H \cdot H'$ removendo uma cópia do q
    para $i \leftarrow 2$ até h' faça
      $h \leftarrow h + 1 \quad H[h] \leftarrow H'[i]$
devolva (H, h)

Consumo de tempo: $T(n) = T(n_d) + T(n_e) + \Theta(n)$,
onde $n = r - p + 1$, $n_d = r - q + 1$ e $n_e = q - p' + 1$.

Miolo recursivo do Quickhull

```
QuickHullRec (X, Y, p, r)
se p = r − 1    ▷ há exatamente dois pontos na coleção
    então h ← 2    H[1] ← r    H[2] ← p
    senão (p′, q) ← Particione(X, Y, p, r)
          (H, h) ← QuickHullRec(X, Y, q, r)
          (H′, h′) ← QuickHullRec(X, Y, p′, q)
          ▷ H ← H · H′ removendo uma cópia do q
          para i ← 2 até h′ faça
              h ← h + 1    H[h] ← H′[i]
devolva (H, h)
```

Consumo de tempo: $T(n) = T(n_d) + T(n_e) + \Theta(n)$,
onde $n = r - p + 1$, $n_d = r - q + 1$ e $n_e = q - p' + 1$.

Observe que $n_d + n_e \leq n$.
Com isso, podemos mostrar que $T(n) = \mathrm{O}(n^2)$.

Casos degenerados e cota inferior

Como tratar os casos degenerados?

Casos degenerados e cota inferior

Como tratar os casos degenerados?

- Cuidado na escolha dos pontos extremos iniciais.

Casos degenerados e cota inferior

Como tratar os casos degenerados?

- Cuidado na escolha dos pontos extremos iniciais.
- Cuidado na escolha do ponto extremo marrom.

Casos degenerados e cota inferior

Como tratar os casos degenerados?

- Cuidado na escolha dos pontos extremos iniciais.
- Cuidado na escolha do ponto extremo marrom.

Será que existe algoritmo melhor para o fecho convexo?

Casos degenerados e cota inferior

Como tratar os casos degenerados?

- Cuidado na escolha dos pontos extremos iniciais.
- Cuidado na escolha do ponto extremo marrom.

Será que existe algoritmo melhor para o fecho convexo?

Podemos ordenar uma sequência de números
usando um algoritmo para fecho convexo!

Casos degenerados e cota inferior

Como tratar os casos degenerados?

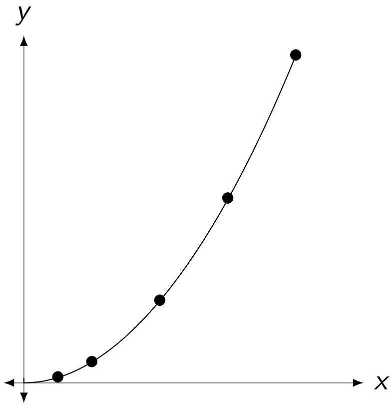
- Cuidado na escolha dos pontos extremos iniciais.
- Cuidado na escolha do ponto extremo marrom.

Será que existe algoritmo melhor para o fecho convexo?

Podemos ordenar uma sequência de números
usando um algoritmo para fecho convexo! Como?

Cota inferior

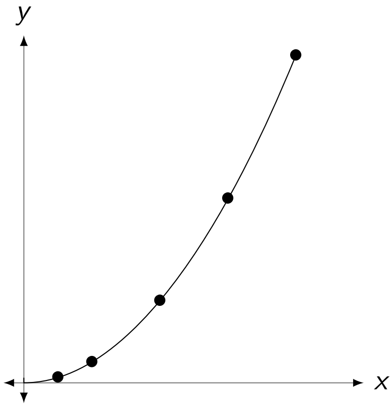
Dados números não-negativos $x_1, x_2, \ldots, x_n$,
considere os pontos $(x_1, x_1^2), (x_2, x_2^2), \ldots, (x_n, x_n^2)$.



Como é o fecho convexo destes pontos?

Cota inferior

Dados números não-negativos $x_1, x_2, \ldots, x_n$,
considere os pontos $(x_1, x_1^2), (x_2, x_2^2), \ldots, (x_n, x_n^2)$.



Como é o fecho convexo destes pontos?

Como obter os números em ordem crescente do fecho convexo?