Geometria Computacional

Cristina G. Fernandes

Departamento de Ciência da Computação do IME-USP

`http://www.ime.usp.br/~cris/`

segundo semestre de 2018

Algoritmo de Hertel e Mehlhorn

Problema: Dado P, determinar uma partição de P por diagonais, em um número mínimo de partes convexas.

Algoritmo de Hertel e Mehlhorn

Problema: Dado P, determinar uma partição de P por diagonais, em um número mínimo de partes convexas.

Seja Φ^* o número mínimo de diagonais que resulta em uma partição convexa de P.

Algoritmo de Hertel e Mehlhorn

Problema: Dado P, determinar uma partição de P por diagonais, em um número mínimo de partes convexas.

Seja Φ^* o número mínimo de diagonais que resulta em uma partição convexa de P.

Algoritmo de aproximação: produz uma partição convexa de P por diagonais com no máximo $\alpha\Phi^*$ diagonais, e consome tempo polinomial.

Um tal algoritmo é chamado de α-aproximação, e α é sua razão de aproximação.

Algoritmo de Hertel e Mehlhorn

Problema: Dado P, determinar uma partição de P por diagonais, em um número mínimo de partes convexas.

Seja Φ^* o número mínimo de diagonais que resulta em uma partição convexa de P.

Algoritmo de aproximação: produz uma partição convexa de P por diagonais com no máximo $\alpha\Phi^*$ diagonais, e consome tempo polinomial.

Um tal algoritmo é chamado de α-aproximação, e α é sua razão de aproximação.

O algoritmo de Hertel e Mehlhorn produz uma partição de P por diagonais, sempre com no máximo $4\Phi^*$ diagonais, ou seja, é uma 4-aproximação.

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Uma diagonal d é essencial para um vértice v se
a remoção de d torna v reflexo (e a partição deixa de ser convexa).

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Uma diagonal d é essencial para um vértice v se
a remoção de d torna v reflexo (e a partição deixa de ser convexa).

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Uma diagonal d é essencial para um vértice v se
a remoção de d torna v reflexo (e a partição deixa de ser convexa).

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Consumo de tempo: linear, usando o algoritmo
de triangulação de Chazelle (que não vimos).

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Uma diagonal d é essencial para um vértice v se
a remoção de d torna v reflexo (e a partição deixa de ser convexa).

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Consumo de tempo: linear, usando o algoritmo
de triangulação de Chazelle (que não vimos).

Por que ele é uma 4-aproximação?

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Lema: Em uma partição convexa por diagonais, existem no máximo duas diagonais essenciais por vértice reflexo.

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Lema: Em uma partição convexa por diagonais, existem
no máximo duas diagonais essenciais por vértice reflexo.

Prova: Feita na aula.

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Lema: Em uma partição convexa por diagonais, existem
no máximo duas diagonais essenciais por vértice reflexo.

Prova: Feita na aula.

Teorema: O algoritmo de Hertel e Mehlhorn é uma 4-aproximação.

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Lema: Em uma partição convexa por diagonais, existem
no máximo duas diagonais essenciais por vértice reflexo.

Prova: Feita na aula.

Teorema: O algoritmo de Hertel e Mehlhorn é uma 4-aproximação.

Prova: Feita na aula.

Partição convexa ótima

Greene apresentou o primeiro algoritmo que
encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $\mathrm{O}(r^2 n^2) = \mathrm{O}(n^4)$, onde n é o número de vértices e r é o número
de vértices reflexos do polígono.

Partição convexa ótima

Greene apresentou o primeiro algoritmo que
encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $\mathrm{O}(r^2 n^2) = \mathrm{O}(n^4)$, onde n é o número de vértices e r é o número
de vértices reflexos do polígono.

Posteriormente, Keil propôs um outro algoritmo que
encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $\mathrm{O}(r^2 n \lg n)$.

Partição convexa ótima

Greene apresentou o primeiro algoritmo que
encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $\mathrm{O}(r^2 n^2) = \mathrm{O}(n^4)$, onde n é o número de vértices e r é o número
de vértices reflexos do polígono.

Posteriormente, Keil propôs um outro algoritmo que
encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $\mathrm{O}(r^2 n \lg n)$.

Os dois algoritmos são de programação dinâmica.

Partição convexa ótima

Greene apresentou o primeiro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $\mathrm{O}(r^2 n^2) = \mathrm{O}(n^4)$, onde n é o número de vértices e r é o número de vértices reflexos do polígono.

Posteriormente, Keil propôs um outro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $\mathrm{O}(r^2 n \lg n)$.

Os dois algoritmos são de programação dinâmica.

Se a partição é por segmentos, o problema fica mais complicado. Há um algoritmo de Chazelle que consome tempo $\mathrm{O}(n + r^3) = \mathrm{O}(n^3)$ para este caso.

Ponto q está no polígono?

P convexo

Ponto q está no polígono?

P convexo

Algoritmo trivial:
verifique se q está à esquerda de todas as arestas de P

Ponto q está no polígono?

P convexo

Algoritmo trivial:
verifique se q está à esquerda de todas as arestas de P

Complexidade: linear

Ponto q está no polígono?

P convexo

Algoritmo trivial:
verifique se q está à esquerda de todas as arestas de P

Complexidade: linear

Algo mais rápido?

Ponto q está no polígono?

P convexo

Algoritmo trivial:
verifique se q está à esquerda de todas as arestas de P

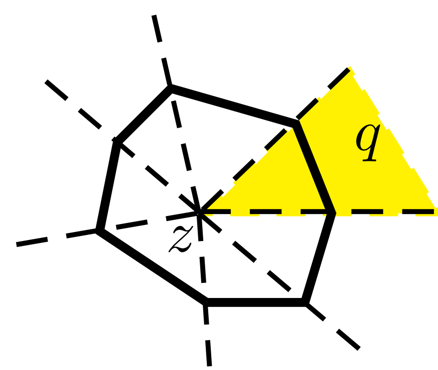
Complexidade: linear

Algo mais rápido?

Várias *queries*:
pré-processamento linear e
queries usando busca binária!

Procure a fatia onde q está.

q está à esquerda ou à direita
da aresta na fatia?



Ponto q está no polígono?

E se P não for convexo?

Ponto q está no polígono?

E se P não for convexo?

Os dois algoritmos anteriores funcionam para alguns polígonos não convexos.

Quais?

Ponto q está no polígono?

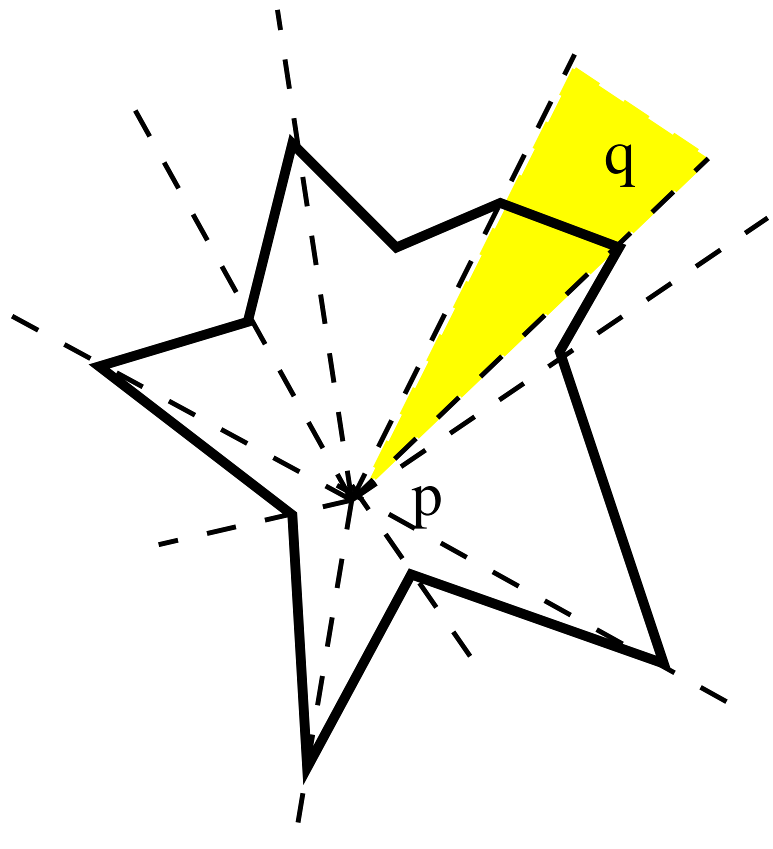
E se P não for convexo?

Os dois algoritmos anteriores funcionam para alguns polígonos não convexos.

Quais?

Polígonos estrela
(star polygon)

P tem um ponto p que enxerga todos os outros pontos de P.



Polígonos arbitrários

Problema: Dados P e q, decidir se q está ou não em P.

Polígonos arbitrários

Problema: Dados P e q, decidir se q está ou não em P.

Dois algoritmos lineares:

- número de voltas (winding number)
- cruzamentos de um raio (ray crossings)

Polígonos arbitrários

Problema: Dados P e q, decidir se q está ou não em P.

Dois algoritmos lineares:

- número de voltas (winding number)
- cruzamentos de um raio (ray crossings)

Ângulo com sinal de $\vec{ab}$ em relação a q:

Winding number

Problema: Dados P e q, decidir se q está ou não em P.

Winding number

Problema: Dados P e q, decidir se q está ou não em P.

Calcule a soma dos ângulos com sinal
de cada aresta de δP em relação a q.

Winding number

Problema: Dados P e q, decidir se q está ou não em P.

Calcule a soma dos ângulos com sinal
de cada aresta de δP em relação a q.

Esse número é ou zero ou 2π.

Winding number

Problema: Dados P e q, decidir se q está ou não em P.

Calcule a soma dos ângulos com sinal
de cada aresta de δP em relação a q.

Esse número é ou zero ou 2π.

Winding number: essa soma dividida por 2π

Winding number

Problema: Dados P e q, decidir se q está ou não em P.

Calcule a soma dos ângulos com sinal
de cada aresta de δP em relação a q.

Esse número é ou zero ou 2π.

Winding number: essa soma dividida por 2π

Se o winding number é zero, q não está em P.
Senão q está em P.

Ray crossings

Problema: Dados P e q, decidir se q está ou não em P.

Ray crossings

Problema: Dados P e q, decidir se q está ou não em P.

R: raio horizontal saindo de q para $x = +\infty$

Quantas vezes R cruza δP?

Ray crossings

Problema: Dados P e q, decidir se q está ou não em P.

R: raio horizontal saindo de q para $x = +\infty$

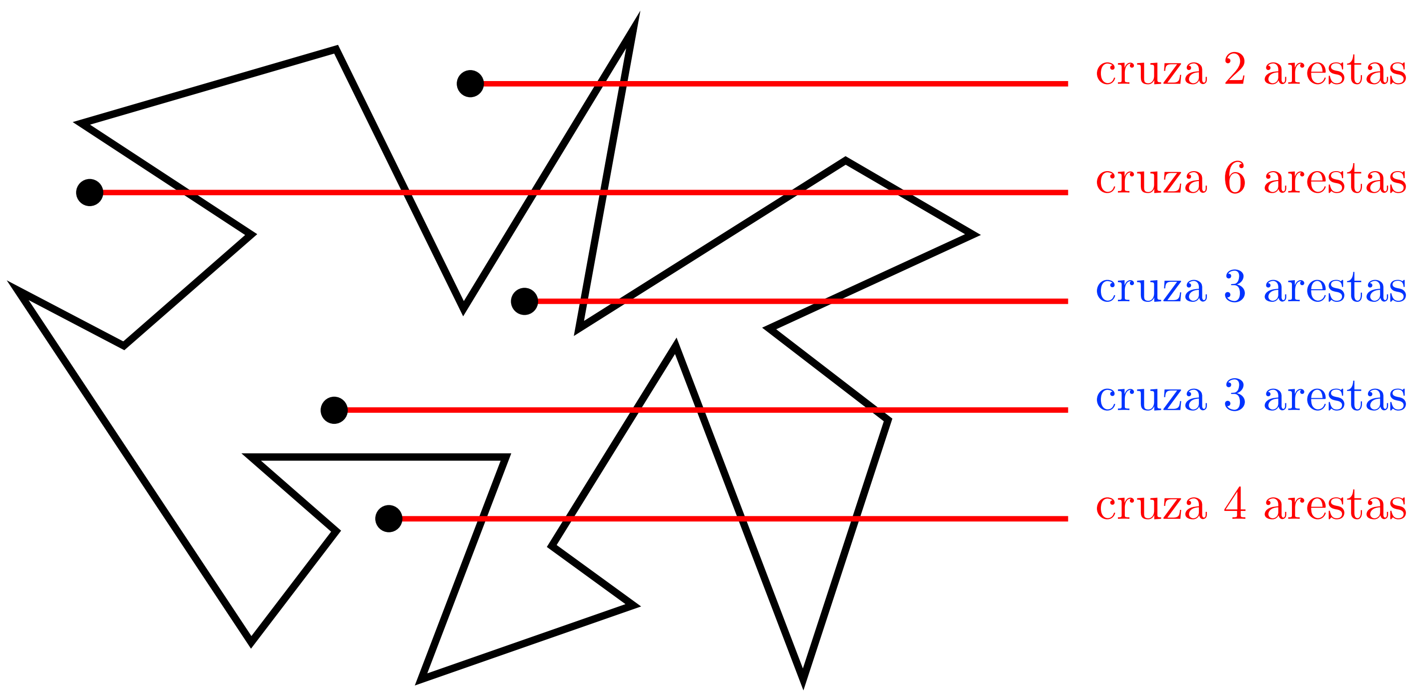
Quantas vezes R cruza δP?
Quantas arestas de δP o raio R cruza?

Ray crossings

Problema: Dados P e q, decidir se q está ou não em P.

R: raio horizontal saindo de q para $x = +\infty$

Quantas vezes R cruza δP?
Quantas arestas de δP o raio R cruza?

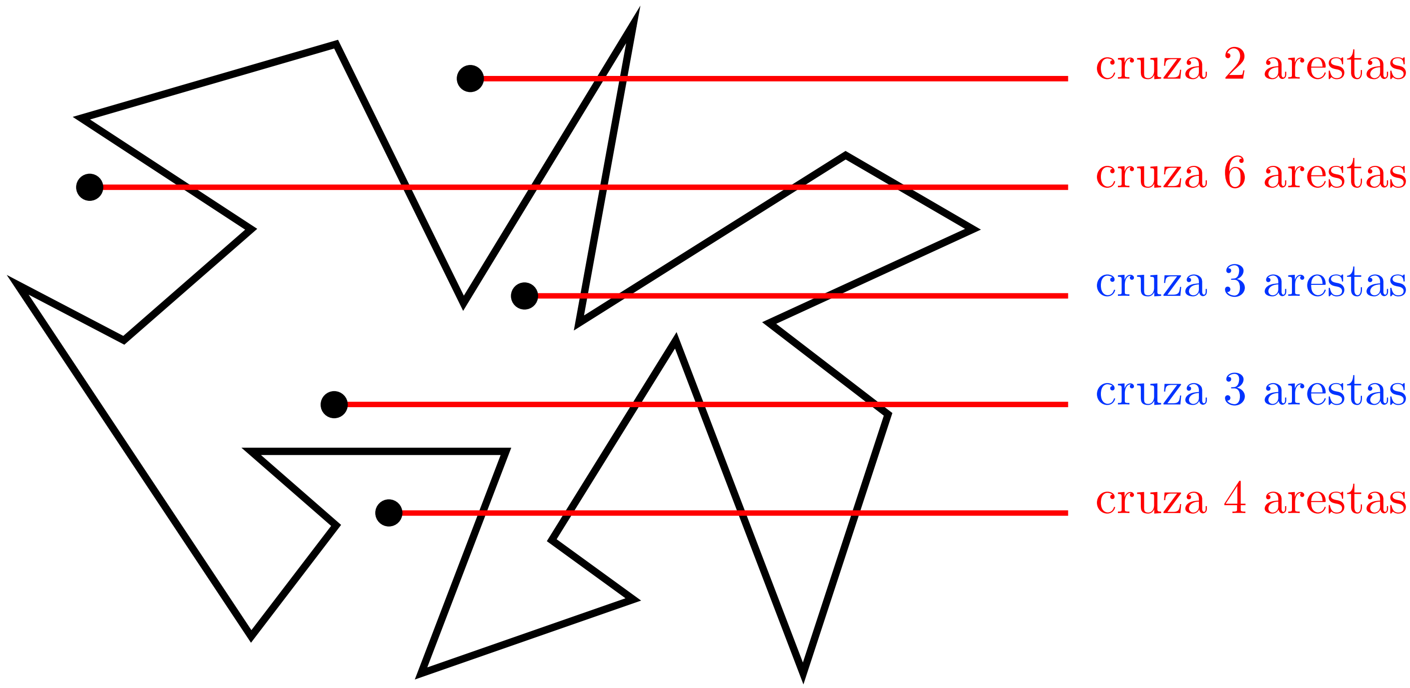


Ray crossings

Problema: Dados P e q, decidir se q está ou não em P.

R: raio horizontal saindo de q para $x = +\infty$

Quantas vezes R cruza δP?
Quantas arestas de δP o raio R cruza?



Se cruza um número par de arestas, q não está em P.
Senão q está em P.

Casos especiais

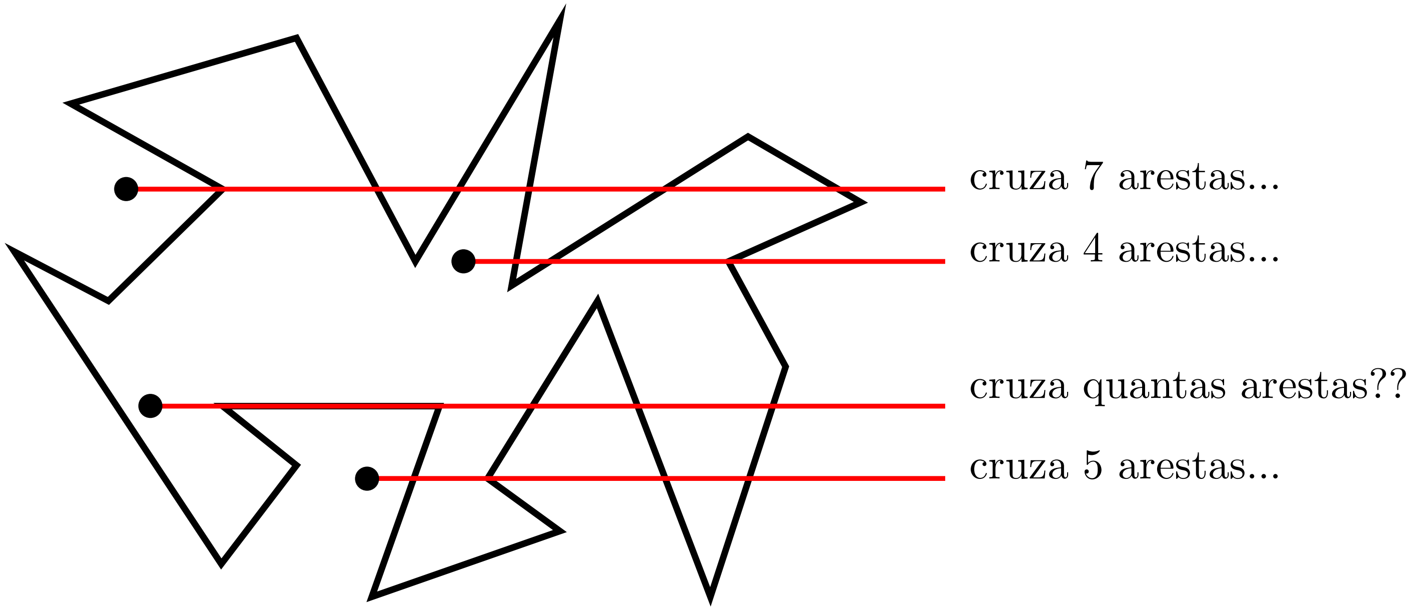
R: raio horizontal saindo de q para $x = +\infty$

Quantas arestas de δP o raio R cruza?

Casos especiais

R: raio horizontal saindo de q para $x = +\infty$

Quantas arestas de δP o raio R cruza?

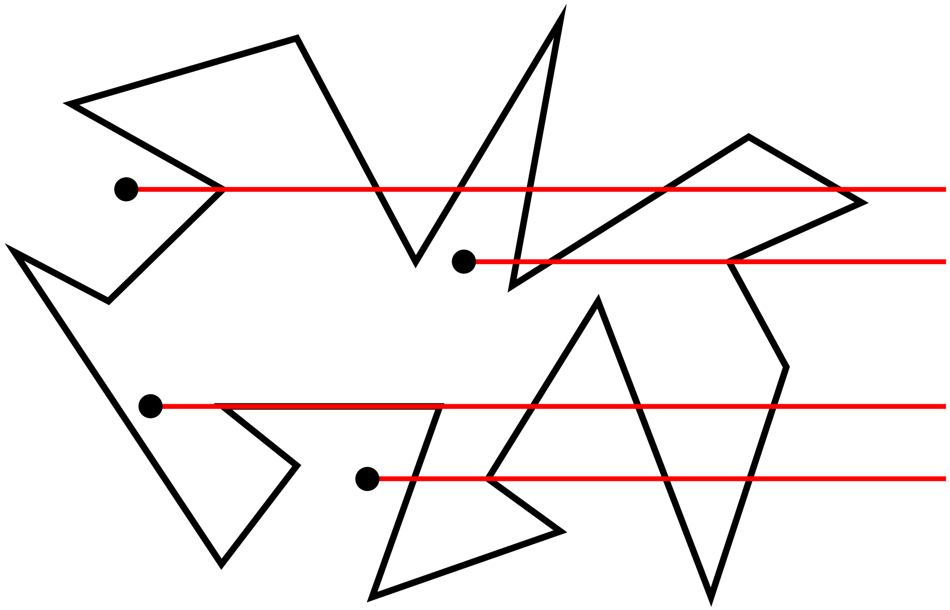


Como tratar destes casos?

Casos especiais

R: raio horizontal saindo de q para $x = +\infty$

Quantas arestas de δP o raio R cruza?



Considere cada aresta
fechada no extremo inferior e aberta no superior!

Cruza: um extremo estritamente acima, outro abaixo.

Primeira versão do algoritmo

Suponha que q é a origem.

Em-Polígono-v0(P, n)

$c \leftarrow 0$
para $i \leftarrow 0$ **até** $n-1$ **faça**
    $j \leftarrow (i + n - 1) \bmod n$ $\triangleright$ vértice $i-1$
    se $(P[i][Y] > 0$ **e** $P[j][Y] \leq 0)$
        ou $(P[j][Y] > 0$ **e** $P[i][Y] \leq 0)$
        então $x \leftarrow (P[i][X] * P[j][Y] - P[j][X] * P[i][Y])$
                    $/(P[j][Y] - P[i][Y])$ $\triangleright$ interseção c/ eixo x
            se $x > 0$
                então $c \leftarrow c + 1$
se c é ímpar
    então devolva dentro
    senão devolva fora

Primeira versão do algoritmo

Suponha que q é a origem.

```
Em-Polígono-v0(P, n)
  c ← 0
  para i ← 0 até n − 1 faça
      j ← (i + n − 1) mod n                              ▷ vértice i − 1
      se (P[i][Y] > 0 e P[j][Y] ≤ 0)
            ou (P[j][Y] > 0 e P[i][Y] ≤ 0)
          então x ← (P[i][X] ∗ P[j][Y] − P[j][X] ∗ P[i][Y])
                        /(P[j][Y] − P[i][Y])          ▷ interseção c/ eixo x
                se x > 0
                    então c ← c + 1
  se c é ímpar
      então devolva dentro
      senão devolva fora
```

Nem sempre funciona quando q está na fronteira de P...

Pontos na fronteira

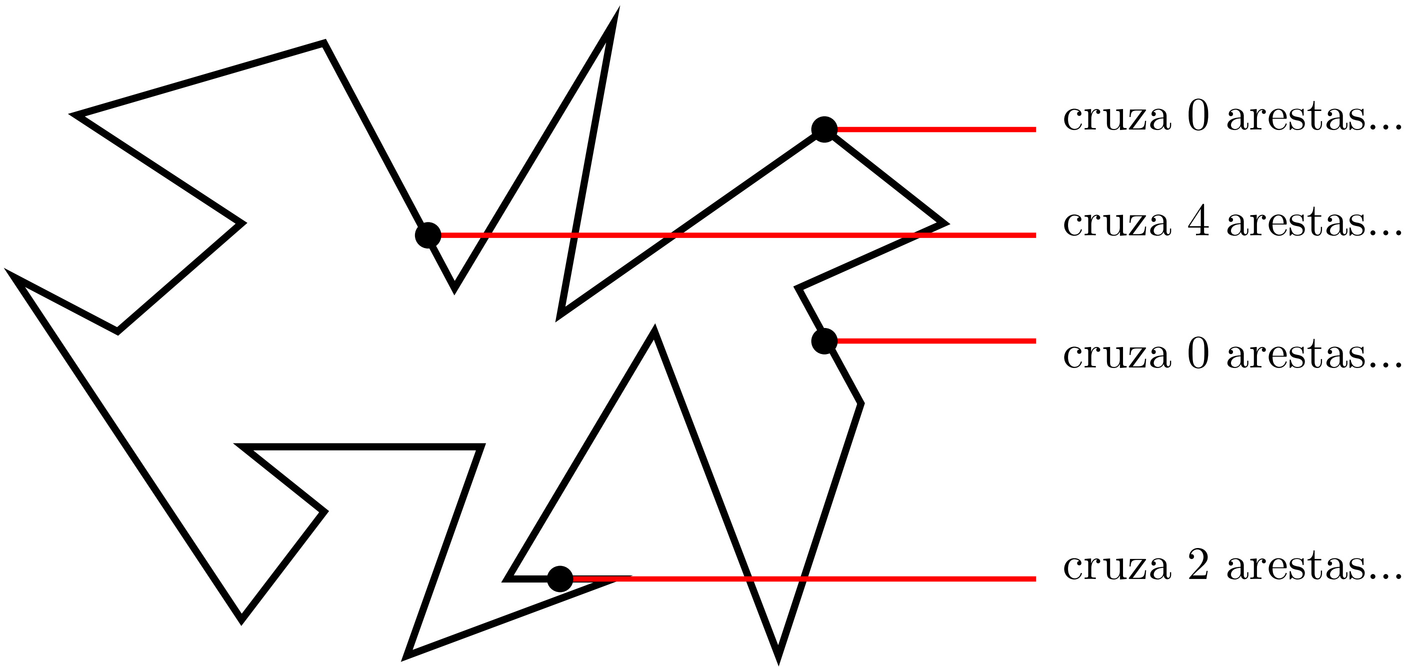
R: raio horizontal saindo de q para $x = +\infty$

Quantas arestas de δP o raio R cruza?

Pontos na fronteira

R: raio horizontal saindo de q para $x = +\infty$

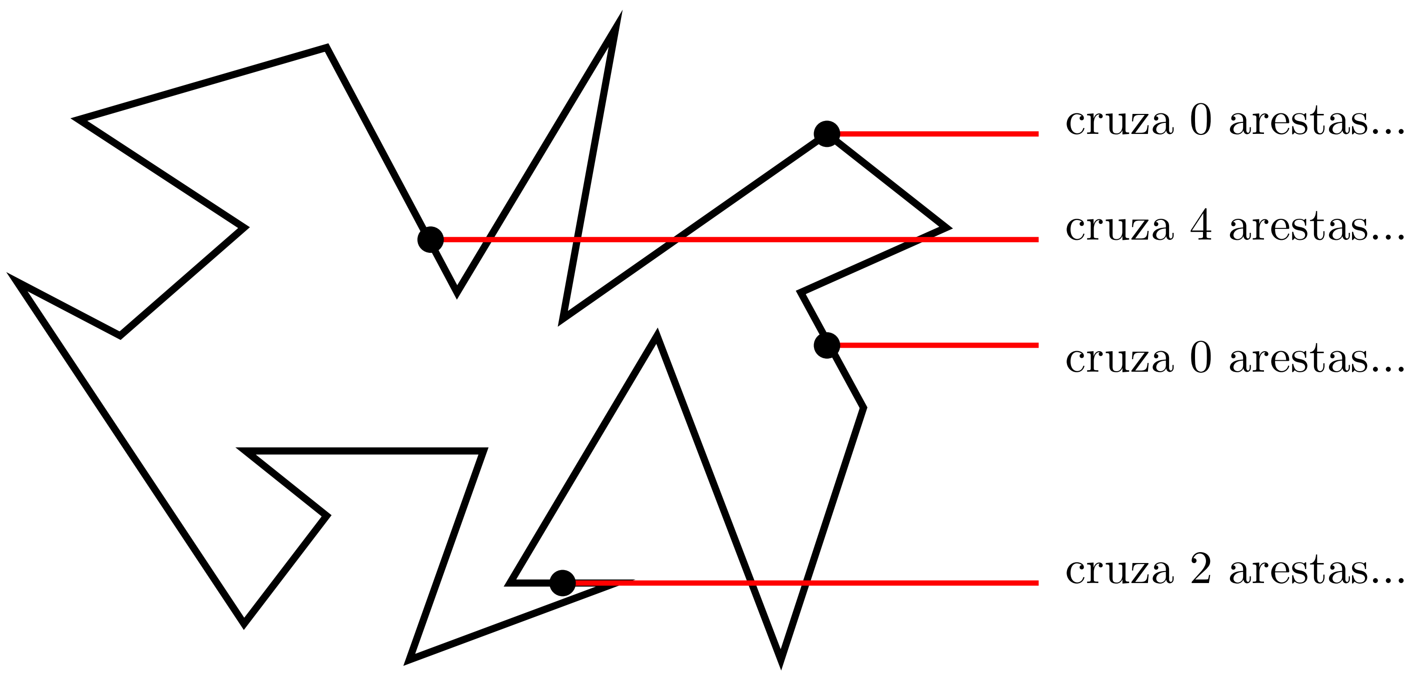
Quantas arestas de δP o raio R cruza?



Pontos na fronteira

R: raio horizontal saindo de q para $x = +\infty$

Quantas arestas de δP o raio R cruza?



Erra em alguns pontos da fronteira,
concluindo que eles estão fora de P...

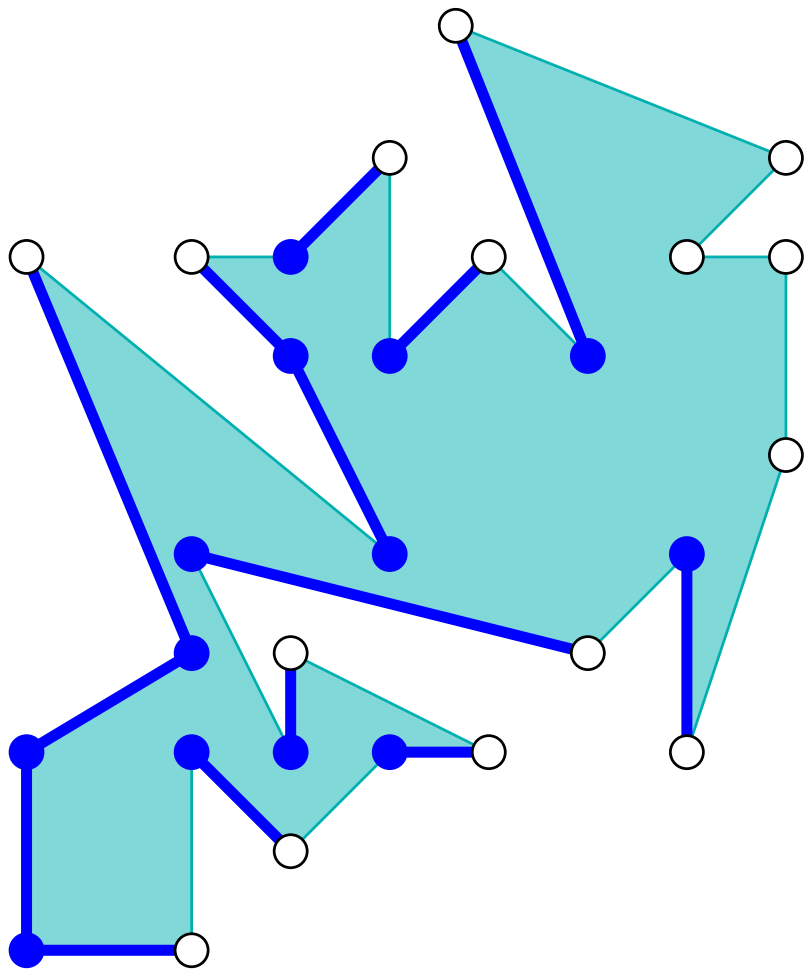
Casos especiais

Se aresta é
fechada no extremo inferior e aberta no superior.

Casos especiais

Se aresta é
fechada no extremo inferior e aberta no superior.

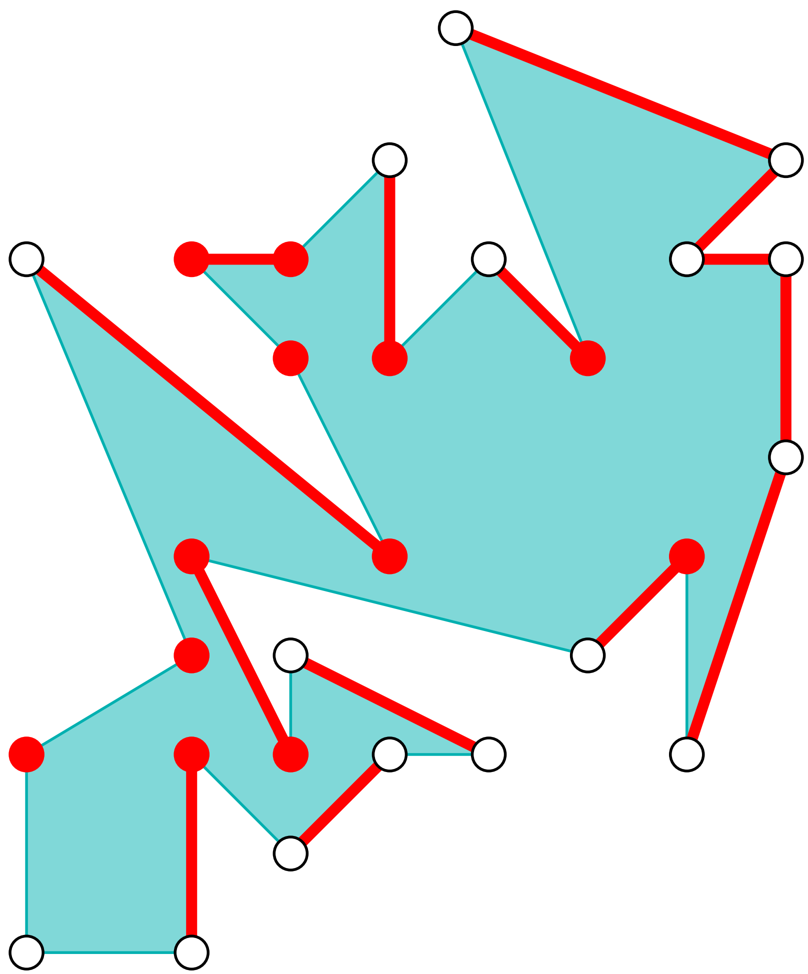
Pontos "em P":



Casos especiais

Agora considere ao contrário: que uma aresta é fechada no extremo superior e aberta no inferior e que R vai para $x = -\infty$.

Pontos "em P":



Casos especiais

Primeira maneira:

acerta no interior das arestas da esquerda e de baixo.

Segunda maneira:

acerta no interior das arestas da direita e de cima.

Casos especiais

Primeira maneira:
acerta no interior das arestas da esquerda e de baixo.

Segunda maneira:
acerta no interior das arestas da direita e de cima.

Interpretando das duas maneiras,
acertamos a resposta no interior de todas as arestas!

Casos especiais

Primeira maneira:
acerta no interior das arestas da esquerda e de baixo.

Segunda maneira:
acerta no interior das arestas da direita e de cima.

Interpretando das duas maneiras,
acertamos a resposta no interior de todas as arestas!

Restam os vértices...

Quando q é um dos vértices de P,
a resposta pode ainda estar errada...

Casos especiais

Primeira maneira:
acerta no interior das arestas da esquerda e de baixo.

Segunda maneira:
acerta no interior das arestas da direita e de cima.

Interpretando das duas maneiras,
acertamos a resposta no interior de todas as arestas!

Restam os vértices...

Quando q é um dos vértices de P,
a resposta pode ainda estar errada...

Faça um teste em separado
para ver se q não é um dos vértices de P.

Segunda versão do algoritmo

Em-Polígono(P, n) $\quad\triangleright$ para facilitar, q é a origem

$c \leftarrow 0$ $\quad d \leftarrow 0$
para $i \leftarrow 0$ **até** $n-1$ **faça**
  se $P[i][X] = 0$ **e** $P[i][Y] = 0$ **então devolva** vértice
  $j \leftarrow (i + n - 1) \bmod n$ $\quad\triangleright$ vértice $i - 1$
  $teste_c \leftarrow (P[i][Y] > 0) \neq (P[j][Y] > 0)$
  $teste_d \leftarrow (P[i][Y] < 0) \neq (P[j][Y] < 0)$
  se $teste_c$ **ou** $teste_d$
    então $x \leftarrow$ intersecção com eixo x
      se $teste_c$ **e** $x > 0$ **então** $c \leftarrow c + 1$
      se $teste_d$ **e** $x < 0$ **então** $d \leftarrow d + 1$
se c e d têm paridade distinta
  então devolva em aresta
se c é ímpar
  então devolva dentro
  senão devolva fora