Geometria Computacional

Polígonos

Prof. Edson Alves

2019

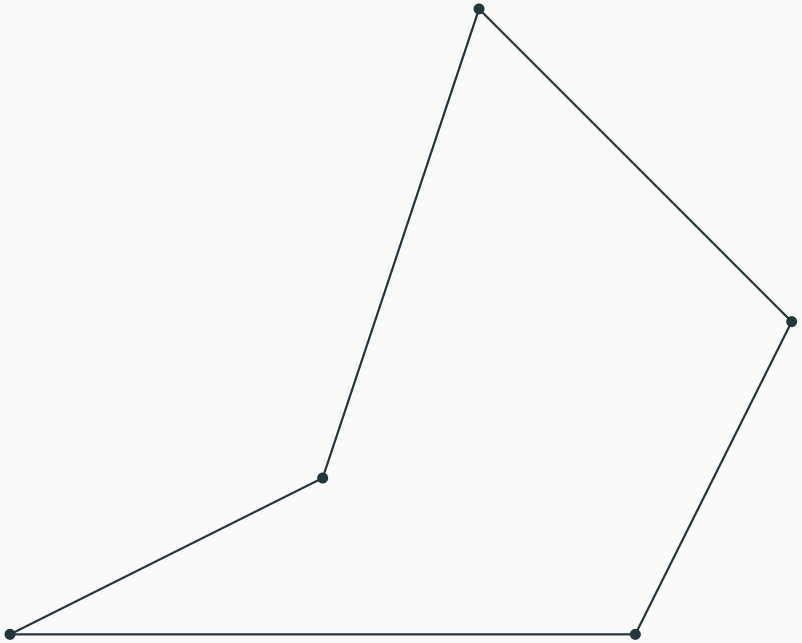
Faculdade UnB Gama

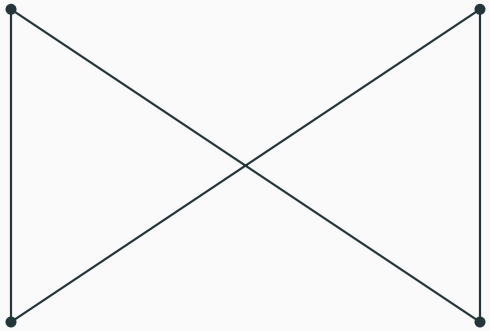
Sumário

Definição

Definição de polígono

- Polígonos são figuras planas delimitadas por caminhos fechados (o vértice de partida é o vértice de chegada), compostos por segmentos de retas que une vértices consecutivos
- Os segmentos que unem os vértices são denominados arestas
- Embora alguns polígonos especiais (triângulos, quadriláteros) possam ter tratamento especial, os algoritmos de polígonos podem ser aplicados igualmente a estes entes geométricos





Representação de polígonos

- A representação mais comum de um polígono é a listagem de seus vértices, sendo que as arestas ficam subentendidas (há sempre uma aresta unindo dois vértice consecutivos)
- Para facilitar a implementação de algumas rotinas, pode ser conveniente inserir, ao final da lista, o ponto de partida
- É preciso tomar cuidado: ao fazer isso, o número de vértices do polígono passa a ser o número de elementos da lista subtraído de uma unidade

```
template<typename T>
using Polygon = vector<Point<T>>;
```

- Esta implementação é a mais compacta possível, mas requer atenção a questão do número de vértices, conforme já comentado
- Uma implementação mais extensa evita os problemas já mencionados

Exemplo de implementação de um polígono em C++

```cpp
#include <bits/stdc++.h>

using namespace std;

template<typename T>
struct Point { T x, y; };

template<typename T>
class Polygon {
private:
    vector<Point<T>> vs;
    int n;

public:
    // O parâmetro deve conter os n vértices do polígono
    Polygon(const vector<Point<T>>& ps) : vs(ps), n(vs.size())
    {
        vs.push_back(vs.front());
    }

```

Polígonos côncavos e convexos

- Um polígono é dito convexo se, para quaisquer dois pontos P e Q localizados no interior do polígono, o segmento de reta PQ não intercepta nenhuma das arestas do polígono
- Caso contrário, o polígono é dito côncavo
- É possível determinar se um polígono é ou não convexo sem recorrer à busca completa isto é, testar todos os possíveis pares de pontos interiores ao polígono
- A orientação D entre pontos e reta pode ser utilizada para tal fim
- Basta checar se, para quaisquer três pontos consecutivos do polígono, eles tem a mesma orientação: ou sempre a esquerda, ou sempre à direita

Implementação da rotina de verificação de convexidade

```
21 private:
22     T D(const Point<T>& P, const Point<T>& Q, const Point<T>& R) const
23     {
24         return (P.x * Q.y + P.y * R.x + Q.x * R.y) -
25             (R.x * Q.y + R.y * P.x + Q.x * P.y);
26     }
27
28 public:
29     bool convex() const {
30         // Um polígono deve ter, no minimo, 3 vértices
31         if (n < 3) return false;
32
33         int P = 0, N = 0, Z = 0, M = vs.size();
34
35         for (int i = 0; i < n; ++i) {
36             auto d = D(vs[i], vs[(i + 1) % M], vs[(i + 2) % M]);
37             d ? (d > 0 ? ++P : ++N) : ++Z;
38         }
39
40         return not ((P and N) or (P == 0 and N == 0));
41     }
```

Referências

1. **HALIM**, Felix; **HALIM**, Steve. *Competitive Programming 3*, 2010.
2. **LAAKSONEN**, Antti. *Competitive Programmer's Handbook*, 2018.
3. **De BERG**, Mark; **CHEONG**, Otfried. *Computational Geometry: Algorithms and Applications*, 2008.
4. David E. Joyce. *Euclid's Elements*. Acesso em 15/02/2019[1]
5. Wikipédia. *Geometria Euclidiana*. Acesso em 15/02/2019[2].

[1] https://mathcs.clarku.edu/ djoyce/elements/bookI/defI1.html
[2] https://pt.wikipedia.org/wiki/Geometria_euclidiana