MAC 747 GEOMETRIA COMPUTACIONAL

LISTA DE EXERCÍCIOS 5

1. Instruções

Entregue os exercícios abaixo até às 17 horas da sexta-feira dia 26 de setembro na secretaria do Departamento de Ciência da Computação (sala 256 do Bloco A).

2. Exercícios

1. (Polígono $\Rightarrow$ quadriláteros convexos) Prove ou dê um contra-exemplo: Todo polígono com um número par de vértices pode ser particionado por diagonais em quadriláteros convexos.
2. (Polígono $\Rightarrow$ quadriláteros) Prove ou dê um contra-exemplo: Todo polígono com um número par de vértices pode ser particionado por diagonais em quadriláteros.
3. (Pior caso em número de partes convexas) Encontre um polígono genérico que pode levar ao pior caso do algoritmo de Hertel e Mehlhorn em termos do número de partes convexas produzidas: existe uma triangularização e uma ordem de remoção das diagonais não-essenciais que leva o algoritmo a produzir $2r$ partes convexas.
4. (*Minimum weight triangulation*) Este exercício mostra um exemplo de um algoritmo geométrico que faz decisões ótimas a cada passo, entretanto ele não fornece uma solução que é ótima globalmente. Dado um conjunto de pontos no plano, uma triangularização é um conjunto maximal de segmentos de reta ligando os pontos tal que dois a dois os segmentos não se intersectam "propriamente" (veja Figura 1). O peso de uma triangularização é a soma dos comprimentos dos segmentos. Dado um conjunto de pontos no plano, o *Problema da Triangularização de Peso Mínimo* é o problema de encontrar, entre todas as triangularizações do conjunto de pontos dado, uma cujo peso seja mínimo.
   [A complexidade do Problema da Triangularização de Peso Mínimo está aberta desde 1975 quando foi mencionado por Shamos e Hoey (cf. [5]). Não se conhece algoritmo polinomial para resolver este problema e também não se sabe se o problema é NP-difícil. O melhor algoritmo de tempo polinomial conhecido produz uma triangularização que, no pior caso é $O(\log n)$ vezes o peso de uma triangularização de peso mínimo (cf. [4], veja também [3]).]

Date: 18 de setembro de 1997.

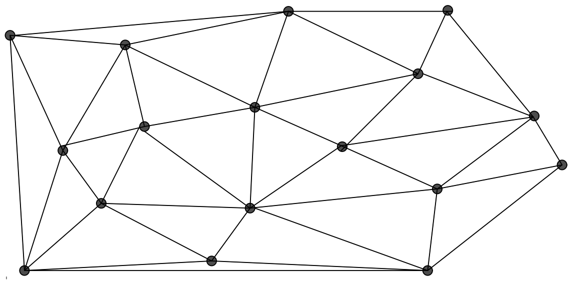

FIGURA 1. Uma triangularização de um conjunto de pontos.

Uma triangularização *gulosa* (*greedy*) de um conjunto de pontos é uma triangularização produzida pelo seguinte algoritmo:

Algoritmo Guloso(S)
`Entrada:` Um conjunto finito S de pontos no plano.
`Saída:` Uma triangularização de S.
$T \leftarrow \emptyset$;
$L \leftarrow$ o conjunto de todos os segmentos ligando pares de pontos de S;
while $L \neq \emptyset$ **do**
  Seja e um segmento em L de menor comprimento;
  Remova e de L;
  if (e não intersecta os segmentos em T) **então**
    Insira e em T;
endwhile

O Algoritmo Guloso escolhe uma aresta de comprimento mínimo em cada passo. Contrua um instância S para o algoritmo tal que S tem 5 pontos e a triangularização gulosa de S não é uma triangularização de peso mínimo. Para isso:
(a) Forneça as coordenadas dos pontos;
(b) Mostre uma triangularização de peso mínimo de S;
(c) Mostre uma triangularização gulosa de S; e
(d) Calcule os pesos de ambas as triangularizações.
[Demnostra-se que a triangularização gulosa produz no pior caso uma triangularização que é $\theta(n)$ vezes mais pesada que uma triangularização ótima.

O Capítulo 16 de [1] apresenta um algoritmo que encontra uma triangularização ótima de um polígono convexo através de programação dinâmica.

Sobre outros problemas de triangularização consulte o Capítulo 4, Seção 6 de [2].]

REFERÊNCIAS

1. T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to algorithms*, The MIT Electrical Engineering and Computer Science Series, The MIT Press, McGraw-Hill Book Company, 1990, QA758 C811i.

2. L.H. Figueiredo and P.C.P. Carvalho, *Introdução à geometria computacional*, 18º Colóquio Brasileiro de Matemática, IMPA, 1991, QA758 F475i.
3. L.S. Heath and S.V. Pemmaraju, *New results for the minumum weight triangulation problem*, Algorithmica **12** (1994), 533–552.
4. D.A. Plaisted and J. Hong, *A heuristic triangulation algorithm*, Journal of Algorithms **8** (1987), 405–437.
5. M.I. Shamos and D. Hoey, *Closest point problems*, Proc. 16th Annual IEEE Symposium in Foundations of Computer Science, 1975, pp. 151–162.