MAC0331 GEOMETRIA COMPUTACIONAL

LISTA DE EXERCÍCIOS 5

1. Instruções

Entregue os exercícios 1,2, 5 e 6 na aula do dia 17 de outubro ou até às 17:00 horas na secretaria do DCC (sala 256 do bloco A).

2. Exercícios

1. (Exercício 1.6.8.2 de [4]) Qual a complexidade de tempo do Algoritmo `Ototomia` quando restringimos a entrada a polígonos convexos?
2. (Ototomia em tempo $O(n^2)$) O algoritmo baseado em remoção de orelhas que foi visto em sala de aula tem complexidade de tempo $O(n^3)$. Nesta questão você desenvolverá um algoritmo de complexidade de tempo $O(n^2)$ que usa a mesma estrutura de dados e a função `Diagonal`.

   Uma chamada para a função `Diagonal` custa tempo $O(n)$. Logo, para obtermos um algoritmo de complexidade de tempo $O(n^2)$, a função `Diagonal` poderá ser chamada apenas $O(n)$ vezes. Imagine que você primeiro determina quais as potenciais diagonais de orelhas são de fato uma diagonal do polígono P, e marque o vértice v_i como 'ORELHA' se $v_{i-1}v_{i+1}$ é uma diagonal, caso contrário marque o vértice como 'NÃO-ORELHA'. Com isto, já gastamos $O(n^2)$ operações elementares.

   Seja P' o polígono de $n-1$ vértices resultante do polígono P após removermos uma de suas orelhas (digamos) $\triangle(v_{i-1}, v_i, v_{i+1})$. Responda as seguintes perguntas e justifique suas respostas:
   (a) É verdade que se o vértice v_j, $j \neq i$, está marcado como ORELHA em P então v_j continua sendo a ponta de uma orelha em P'?
   (b) Para quais valores de j a marca em v_j poderá ser alterada?
   (c) Quantas novas candidatas a diagonais precisam ser verificadas? (Ou seja, quantas novas chamadas da função `Diagonal` precisam ser feitas?)
   (d) Em quais condições um vértice marcado como NÃO-ORELHA em P deverá ser marcado como ORELHA em P'?
   (e) As marcas ORELHA e NÃO-ORELHA de P' herdadas por P podem ser atualizadas em tempo $O(n)$?

   Se a resposta para a última pergunta é sim então um algoritmo de triangularização de complexidade de tempo $O(n^2)$ pode ser obtido simplesmente mantendo-se as marcas ORELHA e NÃO-ORELHA dos vértices do polígono corrente e escolhendo-se, em cada iteração, uma orelha do polígono corrente para ser removida.

Formule precisamente quais segmentos de P' precisam ser analisados para termos todas as orelhas de P' devidamente marcadas. Prove este lema, escreva o algoritmo resultante e a analise a sua complexidade de tempo.

3. Construa um polígono que é não-monótono em relação a qualquer reta.
4. (Exercício 1.6.8.2 de [4] — Pontas interiores) Contrua um polígono monótono que não seja estritamente monótono e que não tenha pontas interiores. (Logo, não é verdade que se um polígono não tem pontas interiores então ele é estritamente monótono.)
5. (Simulação: triangularização de polígonos monótonos) Simule (de alguma maneira clara) a execução do Algoritmo `Monótono` para triangularização de polígonos y-monótonos com o polígono da Figura 1.

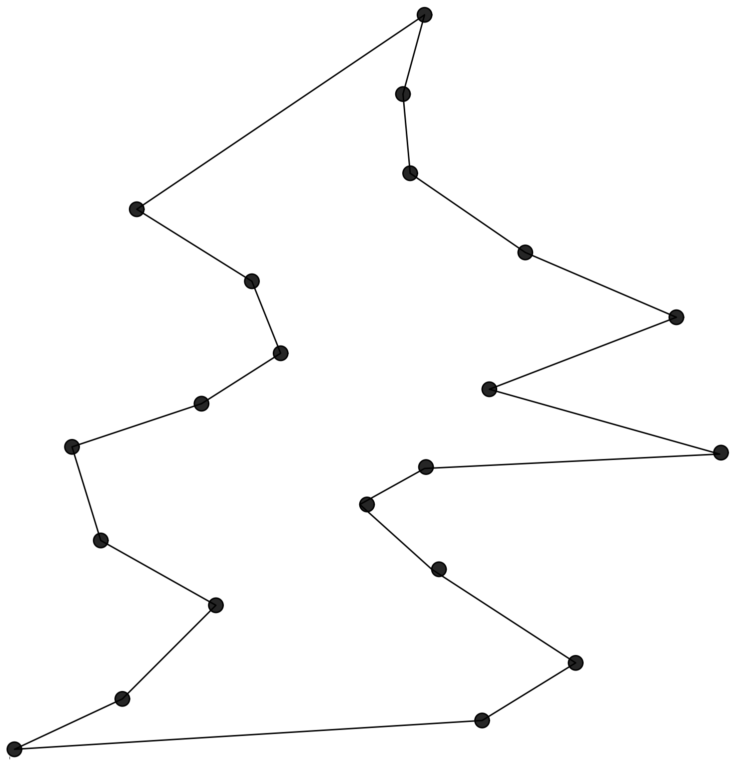

FIGURA 1. Um polígono monótono.

6. (Simulação: partição em polígonos monótonos) Simule (de alguma maneira clara) a execução do Algoritmo por linha-de-varredura `FazMonótono` que particiona um polígono em partes y-monótonos com o polígono da Figura 2.
7. (Exercício 1.6.8.2 de [4] — Monótono em relação a uma única direção) Um polígono pode ser monótono em relação a precisamente um única direção?

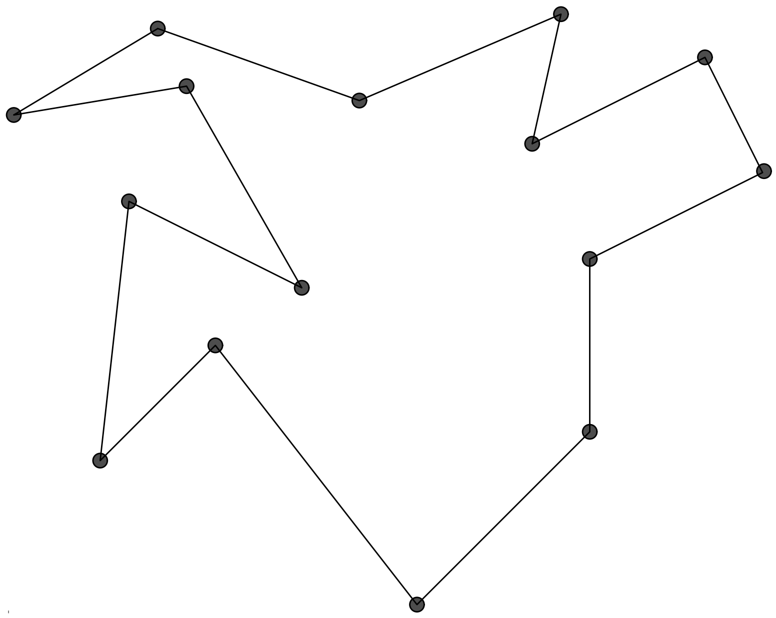

FIGURA 2. Um polígono para ser particionado em polígonos monótonos.

Referências

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

[2] L.H. Figueiredo e P.C.P. Carvalho, *Introdução à geometria computacional*, 18º Colóquio Brasileiro de Matemática, IMPA, 1991, QA758 F475i.

[3] L.S. Heath e S.V. Pemmaraju, New results for the minumum weight triangulation problem, *Algorithmica* **12** (1994), 533–552.

[4] J. O'Rourke, *Computational geometry in C*, Cambridge University Press, Cambridge, 1993, Second Edition, 1998.

[5] D.A. Plaisted e J. Hong, A heuristic triangulation algorithm, *Journal of Algorithms* **8** (1987), 405–437.

[6] M.I. Shamos e D. Hoey, Geometric intersection problems, *Proceedings of the 17th Annual IEEE Symposium on Foundations of Computer Science*, 1976, pp. 208–215.