Geometria Computacional

Departamento de Ciência da Computação – IME/USP

Segundo Semestre de 2007

Lista 6

(1) [O'Rourke 3.4.1.1 — `Embrulho-para-Presente`: melhor caso] Determine o melhor caso do Algoritmo EMBRULHO-PARA-PRESENTE, ou seja, encontre um conjunto de n pontos no plano tal que a complexidade de tempo assintótica do algoritmo seja o menor possível em função de n para tal conjunto. Qual é esta complexidade de tempo?

(2) [O'Rourke 3.4.1.2 — `Embrulho-para-Presente`: melhorias] Durante a execução do algoritmo EMBRULHO-PARA-PRESENTE, algumas vezes é possível determinar alguns pontos que não podem ser vértices do fecho convexo e portanto podem ser eliminado *"on the fly"*. Determine algumas regras para identificar tais pontos. Qual é o conjunto de pontos que exige mais trabalho do seu novo algoritmo?

(3) [O'Rourke 3.4.1.4 — `QuickHull`: pior caso] Construa um conjunto genérico de n pontos que força o Algoritmo QUICKHULL a ter um comportamente quadrático. Por 'genérico', como sempre, queremos dizer uma construção que seja o pior caso para o QUICKHULL para valores arbitrários de n.

(4) [O'Rourke 3.4.1.5 — `QuickHull`: análise do pior caso] Argumente que o QUICHHULL, como o Algoritmo EMBRULHO-PARA-PRESENTE, tem complexidade de tempo *output-sensitive* de $O(nh)$, onde n é o número de pontos e h é o número de pontos na fronteira do fecho convexo.

(5) [O'Rourke 3.4.1.7 — `QuickHull`: complexidade no caso médio] Argumente que o QUICHHULL tem complexidade de tempo $O(n)$ para n pontos escolhidos uniformemente ao acaso no quadrado de dimensão 1.

(6) [O'Rourke 3.8.5.4 — `MergeHull`: *merge* sem ordenação] Se o pré-porcessamento do Algoritmo MERGEHULL não é feito, os fechos convexos podem se intersectar. Descreva um algoritmo que faz o 'merge' de dois fechos convexos arbitrários com k e m vértices em tempo $O(k+m)$. Este seu algoritmo fornece um outro algoritmo por divisão-e-conquista de complexidade de tempo $O(n \lg n)$ para construir o fecho convexo de um conjunto de n pontos.

(7) [O'Rourke 3.5.6.1 — `Graham-Scan`: pior caso] Construa um conjunto de pontos para o qual o **enquanto** da linha 6 do algoritmo de Graham faz o maior número de iterações.

(8) Descreva um algoritmo de complexidade de tempo $O(n \lg n)$ que, dado um conjunto S de n pontos no plano, constrói um polígono que tem os pontos em S como vértices.

(9) Modifique o Algoritmo InsertionHull de tal forma que a sua complexidade de tempo seja reduzida para, no pior caso, $O(n \lg n)$.

[**Sugestão.** Em um pré-processamento ordene os pontos de S em relação a x-coordenada de tal forma que $p \notin Q$ em cada passo. Examine os pontos nesta ordem.]

(10) [O'Rourke 3.9.1.1 — fecho convexo de polígonos monótonos] Descreva um algoritmo que constrói o fecho convexo de um polígono monótono em tempo linear.

(11) [O'Rourke 3.9.1.2 — fecho convexo de um polígono arbitrário] Descreva um algoritmo que constrói o fecho convexo de um polígono arbitrário em tempo linear. Este execício é *tricky* e vários algoritmos publicados para este problema estavam errados.

[**Observação.** A cota inferior de $\Omega(n \log n)$ é para um conjunto de pontos sem estrutura alguma, não para os pontos que são os vértices de um polígono. Os vértices de um polígono são dados na ordem em que eles ocorrem ao percorrermos a fronteira do polígono em sentido anti-horário.]

(12) [Preparata & Shamos 3.6 — Keil-Kirkpatrick] Seja S um conjunto de n pontos no plano com coordenadas inteiras entre 1 e n^d, onde d é uma constante. Mostre que o casco convexo de S pode ser obtido em tempo $O(n)$.

(13) [Preparata & Shamos 3.1 — polígono simples] Dados n pontos no plano, construir um polígono que os tenha como seus vértices.

(a) Mostre uma cota inferior de $\Omega(n \lg n)$ para esse problema.

(b) Projete um algoritmo $O(n \lg n)$ para ele.

Dica: *inspire-se no algoritmo de Graham.*