Geometria Computacional

Departamento de Ciência da Computação – IME/USP

Primeiro Semestre de 2022

Lista 10

1. Ajuste o MergeHull para que funcione sem a hipótese simplificadora, ou seja, para que funcione mesmo que a coleção de pontos dada tenha vários pontos colineares e vários pontos com a mesma Y-coordenada. Teste a sua adaptação do MergeHull com uma entrada que consiste de n pontos colineares.

2. [O'Rourke 3.8.5.4 – MergeHull: *merge* sem ordenação] Se o pré-processamento 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.

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

4. [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 esse problema.
   Dica: *inspire-se no algoritmo de Graham.*

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

6. [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 exercício é delicado. Vários algoritmos publicados para este problema estavam errados.)

   [**Observação.** A cota inferior de $\Omega(n \lg n)$ 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.]

7. [O'Rourke 3.9.3.1 — distância entre polígonos convexos] Sejam A e B dois polígonos convexos. Defina dois tipos de distância entre eles:

$$\begin{aligned} \delta &= \min_{x \in A, y \in B} |x - y|, \\ \Delta &= \max_{x \in A, y \in B} |x - y|, \end{aligned}$$

   onde $|x - y|$ denota a distância no plano entre os pontos x e y. Projete algoritmos para calcular esses dois valores. Você pode assumir que $A \cap B = \emptyset$.

Primeiro, estabeleça alguns lemas geométricos que caracterizem o tipo de pontos que podem atingir a distância mínima ou máxima. É possível que tais pontos estejam no interior de A e B, ou eles devem estar sempre na fronteira? Neste último caso, eles devem ambos ser vértices, pelo menos um deles deve ser vértice, ou podem estar no interior de uma aresta? As respostas a estas questões podem não ser as mesmas para δ e Δ.

Tente encontrar um algoritmo $O(n)$ para Δ e $O(\lg n)$ para δ, onde n é a soma do número de vértices dos dois polígonos. O primeiro algoritmo não é tão difícil, mas o segundo pode ser difícil de projetar. Em particular, ele pode depender de se calcular a mediana de n números em tempo linear.

8. [O'Rourke 3.9.3.2 — diâmetro e largura]

   (a) Defina o diâmetro de um conjunto de pontos $\{p_1, \ldots, p_n\}$ como a distância máxima entre quaisquer dois pontos do conjunto: $\max_{i,j} |p_i - p_j|$. Prove que o diâmetro de um conjunto é atingido por dois vértices do fecho convexo do conjunto.

   (b) Uma *reta de suporte* de um conjunto é uma reta que toca o fecho convexo deste conjunto e deixa todos os pontos do conjunto de um mesmo lado (ou em cima da reta).

   (c) Dois pontos a e b são *antípodas* se eles estão em retas de suporte paralelas (distintas): existem duas retas de suporte paralelas, uma passando por a e outra por b. Projete um algoritmo para listar todos os pares de pontos antípodas de um conjunto de pontos do plano.

   (d) Defina a *largura* como a distância mínima entre duas retas de suporte paralelas (distintas). Projete um algoritmo para calcular a largura de um conjunto de pontos do plano.

9. Utilize a ideia do item (c) do exercício anterior para projetar um algoritmo linear que, a partir do fecho convexo de um conjunto de n pontos do plano, encontra um par de pontos mais distantes do conjunto.