Geometria Computacional

Departamento de Ciência da Computação – IME/USP

Primeiro Semestre de 2020

Lista 7

1. Ajuste todos os algoritmos para fecho convexo vistos em aula para que funcionem sem a hipótese simplificadora, ou seja, para que funcionem mesmo que a coleção de pontos dada tenha vários pontos colineares. Teste cada um dos algoritmos resultantes dos seus ajustes com uma entrada que consiste de n pontos colineares.
2. [O'Rourke 3.4.1.1 – EMBRULHO: melhor caso] Determine o melhor caso do algoritmo EMBRULHO, ou seja, encontre um conjunto de n pontos no plano tal que o consumo de tempo do algoritmo seja o menor possível em função de n para tal conjunto. Quanto é este consumo de tempo em função do n?
3. [O'Rourke 3.4.1.2 – EMBRULHO: melhorias] Durante a execução do algoritmo EMBRULHO, algumas vezes é possível determinar alguns pontos que não podem ser vértices do fecho convexo e portanto podem ser eliminados *"on the fly"*. Determine algumas regras para identificar tais pontos. Qual é o conjunto de pontos que exige mais trabalho do seu novo algoritmo?
4. [O'Rourke 3.5.6.1 – GRAHAM: pior caso] Construa um conjunto de pontos para o qual o **enquanto** da linha 5 do algoritmo de Graham faça o maior número possível de iterações.
5. Implemente o algoritmo INCREMENTAL e veja como é o seu consumo de tempo quando a entrada é um conjunto de n pontos escolhidos uniformemente ao acaso no quadrado $[0\,..\,1] \times [0\,..\,1]$. Experimentalmente qual é o seu consumo de tempo em função de n?
6. [O'Rourke 3.7.1.3 – INCREMENTAL: versão ótima] Modifique o algoritmo INCREMENTAL de tal forma que a sua complexidade de tempo seja reduzida para, no pior caso, $\mathrm{O}(n \lg n)$.

   [**Sugestão.** Em um pré-processamento, ordene os pontos dados pela X-coordenada de tal forma que o teste de pertinência possa ser evitado. Examine os pontos nesta ordem.]
7. [O'Rourke 3.4.1.4 – QUICKHULL: pior caso] Construa um conjunto de n pontos, para um n genérico, para o qual o algoritmo QUICKHULL consome tempo quadrático.
8. [O'Rourke 3.4.1.5 – QUICKHULL: pior caso] Argumente que o QUICKHULL, como o algoritmo EMBRULHO, é *sensível à saída* e consome tempo $\mathrm{O}(nh)$, onde n é o número de pontos e h é o número de pontos na fronteira do fecho convexo.
9. [O'Rourke 3.4.1.6 – QUICKHULL: estudo experimental] Implemente o QUICKHULL e veja como é o seu consumo de tempo quando a entrada é um conjunto de n pontos escolhidos uniformemente ao acaso no quadrado $[0\,..\,1] \times [0\,..\,1]$. Experimentalmente qual é o seu consumo de tempo em função de n?
10. [O'Rourke 3.4.1.7 – QUICKHULL: consumo esperado de tempo] Argumente que o consumo esperado de tempo do QUICKHULL é $\mathrm{O}(n)$ quando este executa com uma instância que consiste de n pontos escolhidos uniformemente ao acaso no quadrado $[0\,..\,1] \times [0\,..\,1]$. *Dica:* A área de um triângulo é metade da área de um paralelogramo com a mesma base.