

# Geometria Computacional

Departamento de Ciência da Computação – IME/USP

*Segundo Semestre de 2014*

## Lista 5

1. Seja  $P$  um polígono convexo cujos vértices estão em um vetor, como sempre no sentido anti-horário. Escreva um algoritmo que decide se um dado ponto  $q$  pertence a  $P$  ou não. O consumo de tempo do seu algoritmo deve ser  $O(\lg n)$ , onde  $n$  é o número de vértices de  $P$ .
2. Seja  $P$  um polígono  $y$ -monótono com  $n$  vértices. Suponha que temos em um vetor  $v[0..n-1]$  os índices dos vértices de  $P$  ordenados por  $y$ -coordenadas. (Podemos obter isso em tempo linear a partir de  $P$ . Você sabe como?) Escreva um algoritmo que decide se um dado ponto  $q$  pertence a  $P$  ou não. A complexidade de tempo do seu algoritmo deve ser  $O(\lg n)$ .
3. Em algumas máquinas (e.g., PCs), divisão de reais pode ser cerca de 20 vezes mais lenta que multiplicação. Aproveite o código da tarefa 4 para fazer alguns testes experimentais. Tenha duas implementações do algoritmo, uma com uma divisão real (linha 8 do algoritmo da aula 13) e outra modificada, evitando a divisão. Cronometre a execução do algoritmo e verifique se há uma diferença significativa na sua máquina.
4. Argumente que, dada uma partição de uma região do plano em polígonos, a primeira versão do algoritmo da aula 13 classifica um ponto em no máximo um dos polígonos.
5. Torne o algoritmo mais rápido, evitando o cálculo de  $x$  (a interseção) quando o segmento está no lado negativo do raio.
6. Modifique o algoritmo para que funcione para um ponto arbitrário  $q$  e não apenas para a origem. Evite fazer a translação do polígono.
7. (a) Modifique o algoritmo para evitar a única operação com reais (cálculo da interseção). Use uma das primitivas vistas.  
(b) Use uma das primitivas para detectar se o ponto está numa das arestas, evitando assim a necessidade da segunda versão do algoritmo.
8. Escreva um algoritmo que determina se uma sequência de  $n$  pontos representa um polígono (ou seja, se a sequência representa uma curva poligonal fechada simples). Você consegue fazer um algoritmo  $O(n \lg n)$  para isso?
9. Descreva um algoritmo  $O(n \lg n)$  para decidir se dois polígonos se intersectam, onde  $n$  é a soma do número de vértices dos dois polígonos.
10. Descreva um algoritmo  $O(n)$  que decide se dois polígonos convexos se intersectam, onde  $n$  é a soma do número de vértices dos dois polígonos.
11. (a) Escreva um algoritmo que usa linha de varredura para pré-processar um dado polígono  $P$  de  $n$  vértices, de forma que, com a estrutura montada, seja possível detectar se um ponto  $q$  pertence ou não a  $P$  em tempo  $O(\lg n)$ .  
(b) Quanto tempo consome seu algoritmo do item anterior?  
(c) Quanto espaço gasta a estrutura montada?  
(d) Escreva o algoritmo que, usando a ED montada pelo algoritmo do item (a), determine se um ponto  $q$  pertence ou não a  $P$  em tempo  $O(\lg n)$ .