MAC 747 GEOMETRIA COMPUTACIONAL

LISTA DE EXERCÍCIOS 3

1. Instruções

Entregar seis (6) entre os oito exercícios abaixo e entregue até às 17 horas da sexta-feira dia 12 de setembro na secretaria do Departamento de Ciência da Computação (sala 256 do Bloco A).

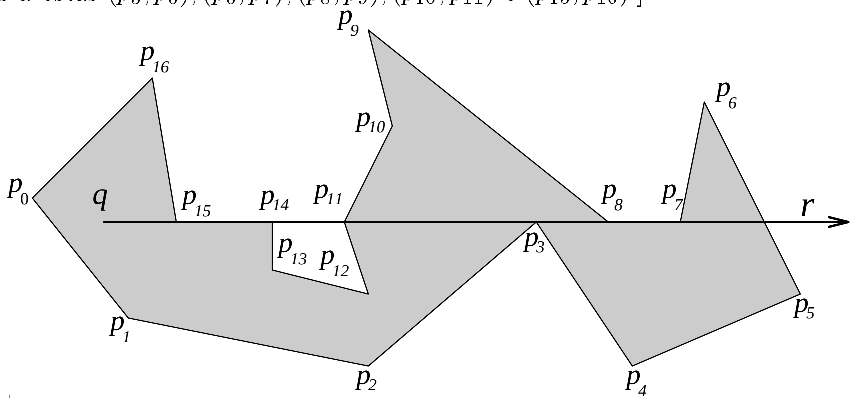
2. Exercícios

1. (Área de um polígono) Seja P um polígono (convexo ou não) com vértices $v_0, v_1, \ldots, v_{n-1}$ listados em ordem anti-horária ao percorrermos a fronteira de P. Mostre que se $v_i = (x_i, y_i)$ $(i = 0, \ldots, n-1)$ então
$$2\mathcal{A}(P) = \sum_{i=0}^{n-1}(x_i y_{i+1} - y_i x_{i+1}),$$
onde $\mathcal{A}(P)$ denota a área do polígono P.
2. (*Integer overflow*) Nos computadores onde variáveis do tipo inteiro podem assumir valores entre -2^{31} e $+2^{31}$, qual é o maior valor que podemos atribuir às coordenadas dos pontos a, b e c para que não ocorra *integer overflow* na função *Area2*?
3. (Exercício 35.1-4 de [1]) O professor Edmorte propôs o seguinte método para determinar se uma seqüência de n pontos $(p_0, p_1, \ldots, p_{n-1})$ são os vértices consecutivos de um polígono convexo. O algoritmo devolve `SIM` se:
   (a) o ponto p_{i+2} está à esquerda do vetor $\overrightarrow{p_i p_{i+1}}$ para $i = 0, 1, \ldots, n-1$;
   ou
   (b) o ponto p_{i+2} está à direita do vetor $\overrightarrow{p_i p_{i+1}}$ para $i = 0, 1, \ldots, n-1$,
   caso contrário o algoritmo devolve `NÃO`. Mostre que apesar deste algoritmo ter complexidade de tempo linear (i.e. $O(n)$) ele não produz sempre uma resposta correta. Modifique o algoritmo do professor Edmorte para que ele sempre forneça uma resposta correta em tempo linear.
4. Note que toda a vez que um *mouse* é 'clicado' dentro de uma janela em um monitor de uma *workstation*, uma instância do *point-in-polygon problem* é resolvida: dado um polígono P e um ponto q, decidir se $q \in P$. Apesar do hardware de algumas máquinas permitirem soluções que evitam geometria, neste exercício e no Exercício 6 nós consideraremos o problema do ponto de vista de geometria computacional.

Mostre como determinar em tempo $\Theta(n)$ se um dado ponto q pertence a um dado polígono convexo P de n vértices.

5. (Exercício 35.1-5 de [1]) Dado um ponto $p_0 = (x_0, y_0)$, o raio horizontal à direita (*right horizontal ray*) de p_0 é o conjunto de pontos $\{p = (x, y) | x \geq x_0 \text{ e } y = y_0\}$, i.e. a semi-reta horizontal à direita de p_0 com p_0 como extremidade. Mostre como decidir se um raio horizontal à direita de um dado ponto p_0 intersecta um segmento p_1p_2 (também dado) em tempo constante (i.e. $O(1)$).
6. (Exercício 35.1-6 de [1]) Uma maneira de determinar se um dado ponto q pertence ou não a um polígono P é tomar um raio r a partir de q e contar o número de intersecções de r com ∂P. O ponto q está no interior de P se o número de 'cruzamentos' de r com ∂P é ímpar ou se q está em ∂P. Mostre como determinar em tempo $\Theta(n)$ se um dado ponto q pertence a um dado polígono P de n vértices.

   [Sugestão. Use o exercício anterior e tenha certeza que o seu algoritmo funciona mesmo quando o raio intersecta vértices do polígono e quando o raio contém arestas do polígono (veja figura abaixo). Um método para eliminar a maioria das dificuldades e exigir que uma aresta e de P seja contada como "cruzada" pelo raio r se uma das pontas de e esteja acima de r é a outra ponta esteja sobre ou abaixo de r. Isto implica que na figura o raio r cruza as arestas $(p_5, p_6), (p_6, p_7), (p_8, p_9), (p_{10}, p_{11})$ e (p_{15}, p_{16}).]



7. (Exercício 35.1-7 de [1]) Mostre como calcular a área de um polígono (não necessariamente convexo) em tempo $O(n)$.
8. (Exercício 35.2-1 de [1]) Mostre que um conjunto de n segmentos que dois a dois se intersectam em no máximo um ponto podem ter $\Theta(n^2)$ intersecções distintas.

Referências

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