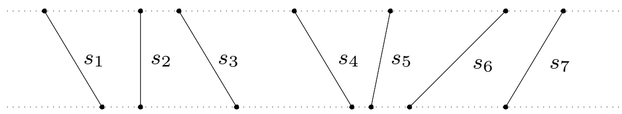
Geometria Computacional

Departamento de Ciência da Computação – IME/USP

Segundo Semestre de 2011

Lista 3

1. [Exercício 2.1 do livro de Berg e outros] Seja S um conjunto de n segmentos de reta disjuntos cujo ponto extremo superior pertence à reta $y = 1$ e o ponto inferior pertence à reta $y = 0$. Estes segmentos particionam a faixa horizontal $[-\infty : \infty] \times [0 : 1]$ em $n + 1$ regiões.



   Descreva um algoritmo que constrói uma árvore de busca binária com os segmentos de S tal que a região contendo um dado ponto possa ser determinada em tempo $\mathrm{O}(\lg n)$. A etapa de construção da árvore deve consumir tempo $\mathrm{O}(n \lg n)$. Também descreva o algoritmo de busca em detalhes.
2. [CLRS 33.2-3] O professor Maqui Sperto sugeriu que modificássemos o algoritmo de Shamos e Hoey (da aula 7, para detectar interseção) de tal forma que, em vez do algoritmo parar logo após detectar uma intersecção, o algoritmo imprimisse os segmentos que se intersectam e continuasse executando a próxima iteração do laço para da linha 3. O professor chama o algoritmo resultante de `Imprime-Segmentos-Intersectantes` e afirma que esse algoritmo imprime todas as interseções, da esquerda para a direita, à medida que elas vão ocorrendo. Mostre que o professor Maqui Sperto está duplamente enganado. Primeiro mostre um exemplo onde a primeira interseção encontrada pelo algoritmo `Imprime-Segmentos-Intersectante` não é a mais à esquerda e depois mostre um exemplo onde o algoritmo não encontra todas as interseções.
3. [CLRS 33.2-4] Mostre um algoritmo que, dado um polígono P com n vértices, decide se P é simples. Seu algoritmo deve consumir tempo $\mathrm{O}(n \lg n)$.
4. [CLRS 33.2-5] Mostre um algoritmo que, dados dois polígonos simples P_1 e P_2, com um total de n vértices, decide se P_1 e P_2 se intersectam. Seu algoritmo deve consumir tempo $\mathrm{O}(n \lg n)$.
5. [CLRS 33.2-6] Um *disco* (ou circulo) é a região do plano limitada por um circunferência. Um disco é dado através das coordenadas do seu centro e do valor do seu raio. Dois discos se intersectam se eles têm algum ponto em comum. Descreva um algoritmo que determine se há dois discos que se intersectem na coleção. Seu algoritmo deve consumir tempo $\mathrm{O}(n \lg n)$.
6. [Exercício 2.11 do livro de Berg e outros] Considere um conjunto S de n circunferências no plano. Descreva um algoritmo de linha de varredura que determina todas as interseções entre as circunferências. Como estamos tratando de circunferências aqui, e não discos, duas circunferências não se intersectam se uma está inteiramente no interior da região limitada pela outra. O seu algoritmo deve consumir tempo $\mathrm{O}((n + I) \lg n)$, onde I é o número de interseções.
7. [Exercício 2.14 do livro de Berg e outros] Considere um conjunto S de n segmentos de reta disjuntos no plano e um ponto p que não pertence a nenhum dos segmentos de S. Desejamos determinar todos os segmentos de S que são visíveis a partir de p, isto é, todos os segmentos de S que contém algum ponto q tal que o segmento aberto pq não intersecta nenhum dos segmentos de S. Descreva um algoritmo para este problema que consuma tempo $\mathrm{O}(n \lg n)$. Pense e use uma semireta com extremo p, rotacionando ao redor de p, como os radares que vocês veêm nos filmes.