MAC 5711 - Análise de Algoritmos

Departamento de Ciência da Computação

Segundo semestre de 2005

Lista 9

1. **Desafio! (Exercício 12.1-4 do CLR)** Desejamos implementar um dicionário usando endereçamento direto em um vetor enoooorme. No início, as entradas do vetor contem lixo e inicializar o vetor inteiro não é recomendável por causa de seu tamanho. Descreva um esquema para implementar um dicionário por endereçamento direto (isto é, a posição i do vetor marca se i está ou não no conjunto) num vetor enorme. Cada objeto lá guardado deve utilizar espaço $O(1)$; as operações `inserção`, `busca` e `remoção` devem consumir tempo $O(1)$ e a inicialização também deve consumir tempo $O(1)$. (*Dica:* use uma pilha cujo tamanho é o número de chaves armazenadas no dicionário para ajudar a determinar se uma entrada do vetor enorme é válida ou não.)

2. **(Exercício 12.3-1 do CLR)** Suponha que desejamos percorrer uma lista ligada de comprimento n onde cada elemento contem uma chave k junto com um valor de hash $h(k)$. Cada chave é uma longa cadeia de caracteres. Como podemos tirar vantagem dos valores de hash quando fazemos uma busca por um elemento com uma dada chave?

3. **(Problema 11-4 do CLRS)** Seja $\mathcal{H}$ uma coleção de funções de hash na qual cada h em $\mathcal{H}$ mapeia o universo U de chaves em $\{0, 1, \ldots, m-1\}$. Dizemos que $\mathcal{H}$ é *k-universal* se, para cada seqüência fixa de k chaves distintas $\langle x^{(1)}, \ldots, x^{(k)}\rangle$ e cada h escolhido aleatoriamente de $\mathcal{H}$, a seqüência $\langle h(x^{(1)}), \ldots, h(x^{(k)})\rangle$ tem a mesma probabilidade de ser qualquer uma das m^k seqüências de comprimento k cujos elementos estão em $\{0, 1, \ldots, m-1\}$.

   (a) Mostre que se $\mathcal{H}$ é 2-universal então $\mathcal{H}$ é universal.

   (b) Seja U o conjunto de n-uplas de valores de $\mathbf{Z}_p$ e seja $B = \mathbf{Z}_p$, onde p é primo. Para cada n-upla $a = \langle a_0, \ldots, a_{n-1}\rangle$ de valores de $\mathbf{Z}_p$ e para cada b em $\mathbf{Z}_p$, defina a função $h_{a,b} : \mathcal{H} \to B$ sobre a n-upla $x = \langle x_1, \ldots, x_k\rangle$ por

   $$h_{a,b}(x) = (\textstyle\sum_{j=0}^{n-1} a_j x_j + b) \bmod p.$$

   Seja $\mathcal{H}$ a família $\{h_{a,b}\}$. Mostre que a coleção $\mathcal{H}_{a,b}$ é 2-universal.

   (c) Suponha que Alice e Bob concordam secretamente sobre uma função de hash $h_{a,b}$ de uma família 2-universal $\mathcal{H}$ de funções de hash. Mais tarde, Alice envia pela internet uma mensagem m a Bob na qual $m \in U$. Ela autentica a mensagem para Bob enviando também $t = h_{a,b}(m)$ e Bob verifica se o par (m, t) que ele recebe satisfaz $t = h_{a,b}(m)$. Suponha que um adversário intercepte (m, t) em trânsito e tente iludir Bob substituindo o par (m, t) por um par (m', t') diferente. Mostre que a probabilidade de o adversário ter sucesso na tentativa de fazer Bob aceitar (m', t') é no máximo $1/p$, independente de quanta capacidade de computação o adversário tenha.

4. Defina *algoritmo eficiente*. Defina *problema de decisão*. Defina *verificador polinomial para* SIM. Defina *verificador polinomial para* NÃO. Defina as classes P, NP e coNP. Dê um exemplo de um problema em cada uma dessas classes, justificando a sua pertinência à classe.

5. Mostre que SAT está em NP. (Essa é a parte fácil do teorema de Cook.)

6. Uma coleção $\mathcal{C}$ de cláusulas sobre um conjunto X de variáveis booleanas é uma *tautologia* se toda atribuição a X satisfaz $\mathcal{C}$. O problema TAUTOLOGIA consiste em, dado X e $\mathcal{C}$, decidir se $\mathcal{C}$ é ou não uma tautologia. O problema TAUTOLOGIA está em NP? Está em coNP? Justifique suas respostas.

7. O problema 2-SAT consiste na restrição de SAT a instâncias X e $\mathcal{C}$ em que toda cláusula de $\mathcal{C}$ tem exatamente dois literais. Mostre que o 2-SAT está em P, ou seja, descreva um algoritmo polinomial que resolva o 2-SAT.

8. Seja $G = (V, E)$ um grafo. Um conjunto $S \subseteq V$ é um *clique* se existe uma aresta entre quaisquer dois vértices em S. O problema CLIQUE consiste no seguinte: dado um grafo G e um inteiro $k \geq 0$, existe um clique em G com k vértices? Mostre que CLIQUE está em NP.

9. Seja $G = (V, E)$ um grafo. Um conjunto $S \subseteq V$ é *independente* se quaisquer dois vértices de S não são adjacentes. Ou seja, não há nenhuma aresta do grafo com as duas pontas em S. O problema IS consiste no seguinte: dado um grafo G e um inteiro $k \geq 0$, existe um conjunto independente em G com k vértices? Mostre que IS é NP-completo.

10. Seja $G = (V, E)$ um grafo. Um conjunto $S \subseteq V$ é uma *cobertura por vértices* de G se toda aresta tem uma ponta em S. O problema VC consiste no seguinte: dado um grafo G e um inteiro $k \geq 0$, existe uma cobertura por vértices em G com k vértices? Mostre que VC é NP-completo.

11. Uma *k-coloração* de um grafo $G = (V, E)$ é uma função f que atribui a cada vértice de V um número em $\{1, \ldots, k\}$ de maneira que dois vértices adjacentes sempre recebem números distintos. O número atribuído a um vértice é usualmente chamado de *cor* do vértice. O problema k-COLORAÇÃO consiste em: dado um grafo G e um inteiro k, decidir se G tem ou não uma k-coloração. Mostre que 2-COLORAÇÃO está em P.

12. Mostre que 3-COLORAÇÃO é NP-completo. *Dica:* Use 3-SAT e os seguintes subgrafos na construção.