MAC 5711 - Análise de Algoritmos

Departamento de Ciência da Computação

Segundo semestre de 2002

Lista 6

1. Os números de Fibonacci podem ser definidos pela seguinte recorrência

$$F_n = \begin{cases} 1 & \text{se } n = 1, 2, \\ F_{n-1} + F_{n-2} & \text{se } n > 2. \end{cases}$$

   Escreva uma função recursiva para calcular F_n. Demonstre que o tempo de execução de tal função é exponencial em n. Explique como você poderia aplicar programação dinâmica para obter um algoritmo mais eficiente. Como é o algoritmo resultante? Quanto tempo ele consome em função do n?

2. **(Exercício 16.1-1 do CLR)** Encontre uma forma ótima de fazer o produto de matrizes cujas dimensões são dadas pela seqüência $\langle 5, 10, 3, 12, 5, 50, 6 \rangle$, ou seja, a primeira matriz tem dimensões 5×10, a segunda matriz tem dimensões 10×3, e assim por diante, até a última matriz, com dimensões 50×6.

3. Mostre que o algoritmo recursivo obtido da recorrência que deduzimos para $m[i, j]$ consome tempo exponencial em $j - i + 1$.

4. **(Exercício 16.1-2 do CLR)** Dê um algoritmo eficiente para **imprimir** uma expressão com parênteses ótima de uma cadeia de matrizes, dada a matriz s computada pelo algoritmo visto em aula. (O seu algoritmo pode ter uma saída do tipo $(A_1(((A_2)(A_3))A_4))$, por exemplo). Analise o seu algoritmo.

5. **(Exercício 16.3-1 do CLR)** Determine uma subseqüência comum de comprimento máximo de $\langle 1, 0, 0, 1, 0, 1, 0, 1 \rangle$ e $\langle 0, 1, 0, 1, 1, 0, 1, 1, 0 \rangle$.

6. **(Exercício 16.3-2 do CLR)** Mostre como reconstruir uma subseqüência comum de comprimento máximo a partir da tabela c completa (gerada pelo algoritmo) e das seqüências originais, sem usar a tabela b.

7. **(Exercício 16.3-4 do CLR)** Mostre como calcular o comprimento de uma subseqüência comum de comprimento máximo usando apenas $2 \min(m, n)$ entradas da tabela c, mais $O(1)$ de espaço adicional. Depois mostre como fazer isso usando $\min(m, n)$ mais $O(1)$ de espaço adicional.

8. **(Exercício 16.3-5 do CLR)** Dê um algoritmo $O(n^2)$ para encontrar uma subseqüência monotonicamente crescente de comprimento máximo de uma seqüência de n números.

9. Use programação dinâmica para escrever um algoritmo que encontre a soma de um segmento de soma máxima de uma lista de n números dados. Analise seu algoritmo e deduza sua complexidade usando notação assintótica.

Entregue no dia 14/10 os exercícios 3, 8 e 9.