Rio de Janeiro, 29 de Maio de 2006.

PROVA 2 DE PROJETO E ANÁLISE DE ALGORITMOS

PROFESSOR: EDUARDO SANY LABER

DURAÇÃO: 2 HORAS

1. Seja um vetor ordenado de n posições $V[1...n]$ contendo números **inteiros distintos**. Exiba um algoritmo eficiente para determinar se existe um inteiro $i \in \{1 \dots, n\}$ tal que $V[i] = i$. Analise a complexidade do algoritmo proposto.

2. Dada duas cadeias $X = x_1x_2...x_n$ e $Y = y_1y_2...y_m$, queremos encontrar o tamanho da maior subcadeia comum entre X e Y, ou seja, o maior valor de k tal que existem índices i e j satisfazendo $x_ix_{i+1}...x_{i+k-1} = y_jy_{j+1}...y_{j+k-1}$. Como exemplo, a maior subcadeia comum entre X=ABABD e Y=AABABBA é BAB. Neste caso, $k = 3, i = 2$ e $j = 3$.

a) Seja $OPT(r, s)$ o tamanho da maior subcadeia comum entre X_r e Y_s, aonde X_r é a subcadeia de X com os r primeiros símbolos de X e Y_s é a subcadeia de Y com os s primeiros símbolos de X. Ache uma equação de recorrência para $OPT(r, s)$. Não esqueça de indicar os casos base.

b) Com base na equação de recorrência, proponha um algoritmo recursivo para calcular $OPT(n, m)$ e analise sua complexidade.

3. Seja $U = \{o_1, ..., o_n\}$ um conjunto de itens, onde o item o_i tem peso inteiro w_i e valor v_i, para $i = 1, \dots, n$. Além disso, seja uma mochila que suporta no máximo W unidades de peso, aonde W é um número inteiro positivo. Definimos $OPT(i, P)$, para $i = 1, \dots, n$ e $0 \leq P \leq W$, como o valor máximo que pode ser obtido colocando um subconjunto de $\{o_1, ..., o_i\}$ na mochila sem exceder o peso P.

a) Explique como seria um procedimento não recursivo para calcular $OPT(n, W)$. Análise a complexidade de tempo e de espaço do procedimento

b) Como seria um procedimento não recursivo para calcular $OPT(n, W)$. O procedimento pode utilizar espaço $O(H)$.

c) Assuma que temos disponível uma tabela $M[0, .., n, 0, .., W]$, onde $M[i, P]$ armazena $OPT(i, P)$. Escreva o pseudo-código de um algoritmo recursivo para obter em tempo $O(n)$ um subconjunto de itens ótimo, ou seja, um com peso menor ou igual a W e com valor $OPT(n, W)$.

4) Encontre um algoritmo eficiente para encontrar a soma dos $n/4$ menores elementos de uma lista contendo n números distintos.

5) O número $B(n)$ de árvores binárias com n nós satisfaz a seguinte equação de recorrência $B(n) = \sum_{i=o}^{n} B(i)B(n-i)$ e $B(1) = 1$ para $n > 2$ e $B(1) = 1$.

a) Projete um algoritmo eficiente que recebe n e devolve o número de árvores binárias de tamanho n.

b) Analise a complexidade de tempo do algoritmo proposto. (Assuma que toda multiplicação e adição tem custo constante).