Rio de Janeiro, 1 de Julho de 2013
PROVA 2 DE PROJETO E ANÁLISE DE ALGORITMOS
PROFESSOR: EDUARDO SANY LABER
DURAÇÃO: 2h:50

1. Sejam X e Y duas strings binárias, cada uma delas com n bits. Para $i = 1, \ldots, n$, seja $x[i]$ o i-ésimo bit de x e $y[j]$ o j-ésimo bit de Y. Uma subsequência comum de X e Y com tamanho k é definida pelos conjuntos de índices $i_1 < i_2 < \ldots < i_k$ e $j_1 < j_2 < \ldots < j_k$ tal que $x[i_m] = y[j_m]$ para $m = 1, \ldots, k$. Como exemplo, se $X = 1010001$ e $Y = 001101$ então $(2, 3, 6, 7)$ e $(1, 4, 5, 6)$ definem uma subsequência comum de tamanho 4 já que $x[2] = y[1] = 0$, $x[3] = y[4] = 1$, $x[6] = y[5] = 0$ e $x[7] = y[6] = 1$.

O objetivo deste problema é encontrar o tamanho da maior subsequência comum entre X e Y. Seja $OPT(i, j)$ o tamanho da maior subsequência comum entre X_i e Y_j, aonde X_i é o prefixo de X que contem os i primeiros bits de X e Y_j é o prefixo de Y que contem os j primeiros bits de Y.

a) (1.0pt) Monte uma equação de recorrência para $OPT(i, j)$. Para isso considere os seguintes casos: (i) x[i] e y[j] **estão associados** na maior subsequência comum entre X_i e Y_j; (ii) x[i] e y[j] **não estão associados** na maior subsequência comum entre X_i e Y_j;

b) (1.0pt) Escreva um algoritmo polinomial e **não-recursivo** para computar $OPT(n, n)$. Analise sua complexidade de tempo e espaço em função de n.

2. Seja $X = \{x_1, x_2, \ldots, x_n\}$ um conjunto de números inteiros que corresponde aos valores das moedas existentes em um dado país. Além disso, sejam V e L números inteiros. Queremos obter um troco de valor V utilizando no máximo L moedas, ou seja, queremos selecionar no máximo L moedas de modo que a soma dos valores das moedas selecionadas seja igual a V. Por exemplo, se $X = \{5, 10\}$ e $L = 7$, é possivel obter um troco de valor $V = 65$ utilizando 7 moedas mas não é possível obter um troco de valor $V = 62$ nem de valor $V = 75$. O objetivo desta questão é projetar a função `Troco` que recebe como entrada um conjunto X de valores de moedas, assim como os valores L e V, e devolve 1 se for possível obter o troco e responde 0, caso contrário.

a) Para $1 \leq i \leq n$, $1 \leq \ell \leq L$ e $1 \leq v \leq V$, defina $Troco(i, \ell, v)$ com valor 1, se for possível obter um troco de valor V utilizando no máximo ℓ moedas do conjunto $\{x_1, \ldots, x_i\}$. Caso contrário, defina $Troco(i, \ell, v)$ com valor 0. Ache uma equação de recorrência para $Troco(i, \ell, v)$.

b) Escreva um algoritmo recursivo para computar $Troco(n, K, V)$ e analise sua complexidade. O algoritmo deve ser polinomial em n, K e V.

3. Seja $A = \{a_1, \ldots, a_n\}$ um conjunto de n números reais positivos. O objetivo desta questão é projetar um algoritmo para encontrar índices i e j, com $1 \leq i \leq j \leq n$, tal que $\prod_{k=i}^{j} a_k$ tem o maior valor possível.

a) Como seria um algoritmo de complexidade $O(n^2)$ para resolver este problema?

b) Como seria um algoritmo de complexidade $T(n)$ que satisfaz $\lim_{n\to\infty} \frac{T(n)}{n^2} = 0$?

4. (2.0pt). Para as equações de recorrência abaixo encontre funções $f(n)$ tal que $T(n) = \Theta(f(n))$

a) $T(n) = n \log n + 2T(n/2)$, para $n > 1$ e $T(n) = 1$ para $n = 1$.

b) $T(n) = n^2 + 4T(n/2)$, para $n > 1$ e $T(n) = 1$ para $n = 1$.

5. Considere o seguinte jogo entre dois jogadores A e B. Inicialmente, uma sequência de n cartas $s_1, \ldots, s_n$ é disposta em uma mesa com a face para cima. Para $i = 1, \ldots, n$, a carta s_i tem valor v_i. O jogo procede em rodadas, alternando jogadas dos jogadores A e B. Em cada rodada o jogador corrente coleta a primeira ou a última carta da sequência remanescente. O objetivo do jogo é maximizar o valor total das cartas coletadas.. Seja $OPT(i,j)$ o valor máximo que um jogador pode coletar quando ele encontra na mesa as cartas $s_i, \ldots, s_j$. Sabemos que $OPT(i,j) = v_i$ se $i = j$ e $OPT(i,j) = \max\{v_i + OPT(i+1,j), v_j + OPT(i,j-1)\}$ se $i < j$.

a) Escreva um algoritmo recursivo para computar $OPT(1,n)$ e analise sua complexidade.

b) Assuma que $OPT(i,j)$ já foi calculado para $1 \leq i \leq j \leq n$ e está armazenado em uma matriz M, ou seja, $M[i,j] = OPT(i,j)$ para $1 \leq i \leq j \leq n$. Escreva um algoritmo para determinar em que ordem as cartas serão retiradas ao longo do jogo se os dois jogadores seguirem a estratégia ótima.