

MAC 338 – Análise de Algoritmos

3ª Lista de Exercícios 1.0 – 6/5/2015 – Entrega 14/5/2015

1. O algoritmo visto em aula para ℓLCS vai da esquerda para a direita. Esta questão consiste em escrever uma variante, da direita para a esquerda. Mais precisamente, suponha dadas sequências x_1^n e y_1^m , e seja

$$rOpt[i, j] = \ell\text{LCS}(x_i^n, y_j^m).$$

- (a) Escreva um esquema recursivo que permita calcular $rOpt[i, j]$ a partir de valores com índices maiores ou iguais. Não se esqueça de ter também uma base de indução, ou seja, como calcular a última linha e a última coluna.
- (b) Escreva uma função em C com o seguinte protótipo

```
int *lLCS(char *x, int n, char *y, int m)
# x[1], ..., x[n], y[1], ..., y[m] como acima, (x[0] e y[0] irrelevantes)
# Devolve um vetor (malocado) L tal que, para todo  $i \geq 1$ ,
#   L[i] = rOpt[i, 1]
```

A função deve rodar em tempo $O(nm)$, e usar $O(n)$ memória. Escreva código que funcione **mesmo**. Entregue código real impresso (se o resto da lista estiver escrito à mão, cole ou grampeie seu código impresso).

2. (baseado em Kleinberg-Tardos, 6.12) Suponha disponível uma coleção de servidores $S_0, S_1, S_2, \dots, S_n$. Um arquivo pode estar replicado em vários deles, sendo dado um custo $c_i > 0$ para replicar o arquivo em S_i ; é garantida a presença de uma cópia em S_0 . Uma requisição de arquivo segue o seguinte protocolo: o arquivo é solicitado a S_i ($i > 0$ é arbitrário, escolhido pelo cliente). Se encontrado, é devolvido, caso contrário a requisição é passada para S_{i-1} , e assim por diante, até encontrar $j \leq i$ tal que S_j contém uma cópia do arquivo. O *custo de requisição em S_i* é definido como sendo $i - j$ (se S_i contém o arquivo, o custo é zero).

Uma *configuração* consiste de uma escolha de um conjunto \mathcal{S} de servidores que conterão cópias do arquivo. O *custo total* de uma configuração consiste da soma do *custo de replicação*, que é a soma dos custos de colocar cópias do arquivo nos servidores em \mathcal{S} e do *custo de requisição* que é a soma dos custos de requisição em S_i , para $i = 1, \dots, n$.

Dê um algoritmo polinomial em n para encontrar uma configuração que minimize o custo total. O algoritmo pode estar em pseudo-código. Deixe claro porque ele funciona e analise a complexidade.