

## MAC0338 - ANÁLISE DE ALGORITMOS

## LISTA 5

exercícios 5 e 16

5. Escreva um algoritmo que, dado  $n$  e um vetor  $v[1..n]$  de inteiros, determina uma subsequência crescente mais longa de  $v$ . Seu algoritmo deve consumir tempo  $O(n^2)$ . Se puder, dê um algoritmo para este problema que consome tempo  $O(n \lg n)$ .

SSCL( $v$ )

```
1  cria vetor aux[1...n]
2  cria vetor resultado[1...n] preenchido com -1
3  aux[1] = 1 // inicialmente a SSLC temporária inicia no primeiro item
4  tamanho = 1
5  para i ← 2 até n faça
6      se v[aux[1]] > v[i] faça // substitui o primeiro item pelo menor
7          aux[1] ← i
8      senão se v[t[tamanho]] < v[i] faça // adiciona um item maior
9          tamanho ← tamanho + 1
10         aux[tamanho] ← i
11         resultado[aux[tamanho]] ← aux[tamanho - 1]
12     senão // procura onde encaixar o item
13         j ← busca(v, aux, tamanho, v[i])
14         aux[j]
15         resultado[aux[j]] ← aux[j - 1]
16     i ← 1
17     cria vetor subseq[1...tamanho]
18     j ← aux[tamanho]
19     enquanto j é diferente de -1 // guarda a subsequência crescente mais longa em
20         subseq[i] ← v[j] // um vetor e devolve
21         j ← resultado[j]
22     i ← i + 1
23     retorna subseq
```

busca(v, auxc, tamanho, item) // busca a maior posição para o item (busca binária)

1 início  $\leftarrow 1$

2 fim  $\leftarrow$  tamanho

3 enquanto início  $\leq$  fim

4     meio  $\leftarrow (\text{início} + \text{fim}) / 2$

5     se  $v[\text{auxc}[\text{meio}]] < \text{item}$  e  $\text{item} \leq v[\text{auxc}[\text{meio} + 1]]$  faça

6         retorne meio + 1

7     senão se  $v[\text{auxc}[\text{meio}]] < \text{item}$  faça

8         início  $\leftarrow$  meio + 1

9     senão fim  $\leftarrow$  meio - 1

16. **PC 111105** (Cortes de tora) Você deve cortar uma tora de madeira em vários pedaços. A empresa mais em conta para fazer isso é a *Analog Cutting Machinery (ACM)*, que cobra de acordo com o comprimento da tora a ser cortada. A máquina de corte deles permite que apenas um corte seja feito por vez.

Se queremos fazer vários cortes, é fácil ver que ordens diferentes destes cortes levam a preços diferentes. Por exemplo, considere uma tora com 10 metros de comprimento, que tem que ser cortada a 2, 4 e 7 metros de uma de suas extremidades. Há várias possibilidades. Podemos primeiramente fazer o corte dos 2 metros, depois dos 4 e depois dos 7. Tal ordem custa  $10+8+6 = 24$ , porque a primeira tora tinha comprimento 10, o que restou tinha 8 metros de comprimento e o último pedaço tinha comprimento 6. Se cortássemos na ordem 4, depois 2, depois 7, pagaríamos  $10 + 4 + 6 = 20$ , que é mais barato.

Seu chefe encomendou um programa que, dado o comprimento  $l$  da tora e  $k$  pontos  $p_1, \dots, p_k$  de corte da tora, encontre o custo mínimo para executar esses cortes na ACM.