

## MC458 - Projeto e Análise de Algoritmos I

### Prova Individual - 28/11/2018

1. Comece pelas questões que você tem mais certeza de saber fazer.
2. Não perca tempo com detalhes menores. Foque no que é relevante para resolver a questão e passe logo para a próxima.
3. Se sobrar tempo, você pode acrescentar mais detalhes.

**Questão 1** (valor 2,5) Construa uma matrix de programação dinâmica  $c$  usada para computar o tamanho de uma subsequência comum mais longa entre as sequências a seguir, e explique o significado do valor em  $c[i, j]$ .

TGCA  
ATC

**Questão 2** (Valor 2,5) Construa uma matrix de programação dinâmica  $M$  usada para encontrar o subconjunto dos itens abaixo de maior valor que caiba numa mochila de capacidade 7 kg:

Item	R\$	kg
1	2	1

Item	R\$	kg
2	1	3

Item	R\$	kg
3	3	2

**Questão 3** (Valor 2,5) Escreva um algoritmo eficiente para o seguinte problema. Sua entrada será uma lista de atividades numeradas de 1 a  $n$ , e dois vetores  $c[1..n]$  e  $f[1..n]$  indicando os momentos iniciais e finais, respectivamente, de cada atividade, medidos em minutos a partir de uma referência fixa no tempo. Deseja-se saber o número máximo de atividades que podem ser alocadas numa mesma sala, sabendo-se que é necessário um intervalo de 15 minutos e trinta segundos entre uma atividade e a próxima, para limpeza da sala.

**Questão 4** (Valor 2,5) Ache um código de Huffman para a seguinte lista de caracteres e suas frequências:

a: 1; b: 1; c: 2; d: 3; e: 5.

Boa sorte!

# Soluções

Questão 1 (valor 2,5)

		A	T	C
	0	0	0	0
T	0	0	1	1
G	0	0	1	1
C	0	0	1	2
A	0	1	1	2

Questão 2 (valor 2,5)

			0	1	2	3	4	5	6	7
v	w	0	0	0	0	0	0	0	0	0
2	1	1	0	2	2	2	2	2	2	2
1	3	2	0	2	2	2	3	3	3	3
3	2	3	0	2	3	5	5	5	6	6

Questão 3 (valor 2,5)

```

function MAXATIVIDADES(c, f)
  ordenar c e f usando f como chave
  n ← length[c]
  A ← {1}                                ▷ A será uma lista ótima de atividades
  i ← 1                                    ▷ i é a atividade mais recentemente selecionada
  for m ← 2 to n do
    if c[m] ≥ f[i] + 15.5 then        ▷ dando tempo para a limpeza
      A ← A ∪ {m}                        ▷ seleciona atividade m
      i ← m
  return size[A]
  
```

Questão 4 (valor 2,5)

A seguir descrevemos a construção da árvore, mostrando o conteúdo da fila de prioridades passo a passo, com os dois menores elementos aparecendo primeiro. Em seguida, exibimos o código de Huffman resultante, usando 0 para o primeiro filho e 1 para o segundo filho.

**Construção da árvore:**

a:1, b:1, c:2, d:3, e:5  
ab:2, c:2, d:3, e:5  
d:3, abc:4, e:5  
e:5, abcd:7  
abcde:12

**Código resultante:**

a: 1100  
b: 1101  
c: 111  
d: 10  
e: 0