

## Soluções

**Unidade Curricular:** 21045 – Estruturas de Dados e Algoritmos Avançados

**Prova:** Época normal, ano letivo de 2014/15, data de 2015/01/28

**Grupo I** [6 valores]

**1.1** [2]

É necessário acrescentar redundância às mensagens para impedir que um intruso que substitua partes do texto cifrado obtenha mensagens válidas, embora desconhecendo o seu significado. Esta estratégia poderia ser utilizada para prejudicar o recetor, que aceitaria as mensagens.

**1.2.1** [2]

Tabela da caixa P	Tabela da caixa S
bits entrada 1036 2547	palavra entrada 0123 4567 89AB CDEF
bits saída 0123 4567	palavra saída B10A ED9C 62F4 8753

Caixa P  
bits 7654 3210  
entrada X 0111 0100  
saída Y 0111 1000

Caixas S  
bits 7654 3210  
entrada Y 0111 1000  
saída Z 1100 0110

**1.2.2** [2]

Partindo das tabelas das caixas P e S, trocando a entrada pela saída (troca de linhas) obtém-se a correspondência entrada/saída das caixas inversas,

caixa P <sup>-1</sup>	caixa S <sup>-1</sup>
bits entrada 0123 4567	palavra entrada B10A ED9C 62F4 8753
bits saída 1036 2547	palavra saída 0123 4567 89AB CDEF

reordenando as colunas de modo a obter a forma padrão convencional para as chaves,

Tabela da caixa P <sup>-1</sup>	Tabela da caixa S <sup>-1</sup>
bits entrada 1042 6537	palavra entrada 0123 4567 89AB CDEF
bits saída 0123 4567	palavra saída 219F BE8D C630 754A

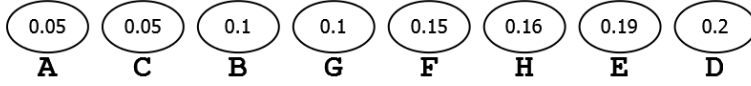
Chave da caixa P<sup>-1</sup>: 1042 6537

Chave da caixa S<sup>-1</sup>: 219F BE8D C630 754A

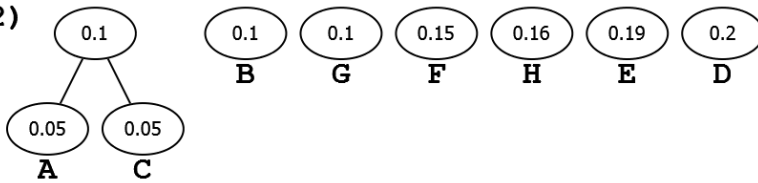
**Grupo II [8 valores]**

**2.1 [3]**

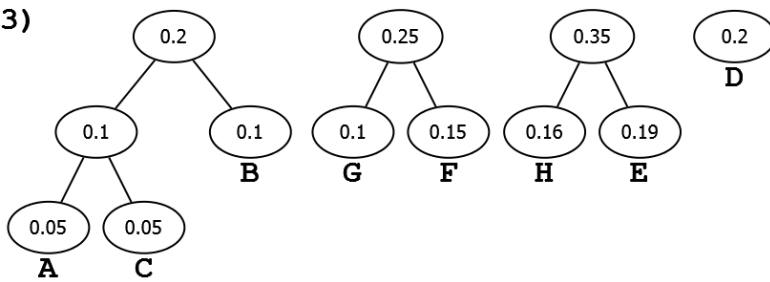
1)



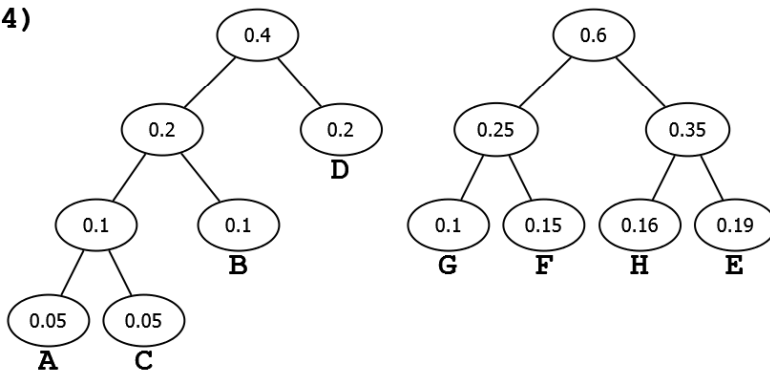
2)



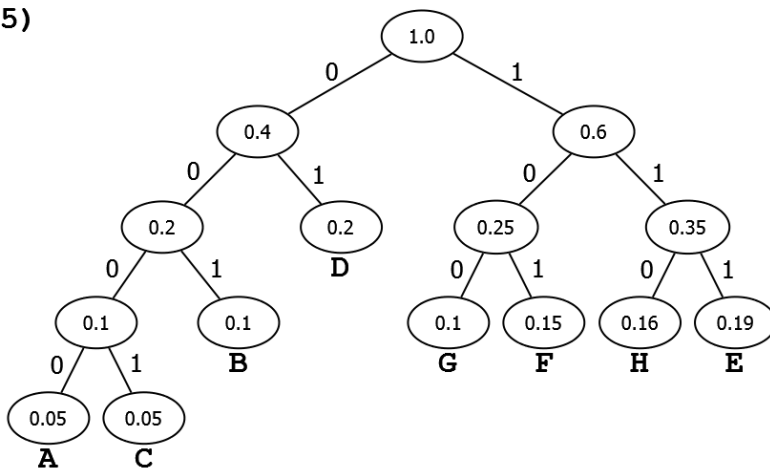
3)



4)



5)



Símbolo	Codificação de Huffman
A	0000
C	0001
B	001
G	100
F	101
H	110
E	111
D	01

## 2.2 [3]

Mensagem= A,B,C,BB,CBA,ABB,ABA,BABABC,CA,BAC,BCCAC,CAB,CABC  
(as vírgulas separam a contribuição de cada tripla)

Codificação LZ77		
Buffer Procura 7←0	Look-ahead buffer (Entrada)	Triplas (Saída)
	A...	(0, 0, A)
A	B...	(0, 0, B)
AB	C...	(0, 0, C)
ABC	BB...	(1, 1, B)
ABCBB	CBA...	(2, 2, A)
ABCBBCBA	ABB...	(7, 2, B)
BBCBAABB	ABA...	(2, 2, A)
BAABBABA	BABABC...	(3, 5, C)
BABABABC	CA...	(0, 1, A)
BABABCCA	BAC...	(5, 2, C)
ABCCABAC	BCCAC...	(6, 4, C)
BACBCCAC	CAB...	(2, 2, B)
BCCACCAB	CABC	(2, 3, C)

$$l_{in} = 37 \times 8 = 296 \text{ bits}$$

$$l_{out} = 13 \times (3 + 3 + 8) = 182 \text{ bits}$$

$$\text{Taxa de compressão} = \frac{296-182}{296} = 38.5\%$$

## 2.3 [2]

RLE é uma técnica em que a ideia base é codificar uma sequência de caracteres (ou símbolos) idênticos pelo par (ch,n) onde ch indica o caracter e n o número de caracteres consecutivos. A técnica pode ter variantes, incluindo o uso de triplas (cm,ch,n) onde cm é uma marca que assinala que se segue a codificação de uma sequência. A técnica RLE só é eficaz em casos em que garantidamente existem muitos caracteres repetidos.

### Grupo III [6 valores]

#### 3.1 [2]

Exemplo de pseudocódigo (com ou sem especificação vértice inicial, a fila é FIFO):

```
BreadthFirstSearch()
  para todos os vertices v
    num(v)=0;
  i=1;
  enquanto existe vertice v com num(v)=0
    num(v)=i++;
    inserir_fila(v);
    enquanto a fila não estiver vazia
      v=retirar_fila();
      para todos os vertices u adjacentes a v
        Se num(u)=0
          num(u)=i++;
          inserir_fila(u);
end
```

Ordem de visita: c, a, d, e, b, g, f, h, i, j.

#### 4.1 [0,5]

Grafo simples, orientado (digrafo), ponderado.

#### 4.2 [0,5]

Matriz de Adjacências									
	a	b	c	d	e	f	g	h	i
a	0	1	0	0	0	0	0	0	0
b	0	0	0	0	1	0	0	0	0
c	0	0	0	1	0	0	1	1	0
d	1	0	0	0	1	0	0	0	1
e	0	0	0	0	0	1	0	0	0
f	0	0	0	0	0	0	0	0	0
g	0	0	0	1	0	0	0	0	0
h	0	0	0	0	0	0	1	0	0
i	0	0	0	0	0	1	0	0	0

#### 4.2 [3]

Ordem das arestas: ab, be, cd, cg, ch, da, de, di, ef, gd, hg, if

Algoritmo de Ford							
Vértice	Iteração						
	0	1	2	3	4	5	
a	$\infty$	3		2	1		
b	$\infty$			4	3	2	
c	0						
d	$\infty$	1	0	-1			
e	$\infty$	5		-1	-2	-3	
f	$\infty$	9	3	2	1		
g	$\infty$	1	0				
h	$\infty$	1					
i	$\infty$	2		1	0		

A distância mais curta entre o vértice  $c$  e o vértice  $e$  é  $-3$ . O caminho é  $c, h, g, d, a, b, e$ .

FIM