

## Soluções

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

**Prova:** Época normal, ano letivo de 2013/14, data de 2014/01/29

### Grupo I [6 valores]

#### 1.1 [2]

É necessário porque um intruso ativo pode inserir mensagens no canal de comunicação que são recebidas e aceites como válidas pelo destinatário. Uma solução pode ser utilizar um tempo de validade, por exemplo de 10 segundos, em conjunto com filtragem de mensagens duplicadas.

#### 1.2.1 [2]

Tabela da caixa P  
bits entrada 4352 1076  
bits saída 0123 4567

Tabela da caixa S  
palavra entrada 0123 4567 89AB CDEF  
palavra saída E986 2D4F 70CB 13A5

Caixa P  
bits 7654 3210  
entrada X 1010 0111  
saída Y 0111 1100

Caixas S  
bits 7654 3210  
entrada Y 0111 1100  
saída Z 1111 0001

#### 1.2.2 [2]

Tabela da caixa  $P^{-1}$   
bits entrada 5431 0276  
bits saída 0123 4567

Tabela da caixa  $S^{-1}$   
palavra entrada 0123 4567 89AB CDEF  
palavra saída 9C4D 6F38 21EB A507

Caixas  $S^{-1}$   
bits 7654 3210  
entrada Z 1100 0101  
saída Y 1010 1111

Caixa  $P^{-1}$   
bits 7654 3210  
entrada Y 1010 1111  
saída X 0111 1101

## Grupo II [8 valores]

### 2.1 [3]

Símbolo	Codificação de Huffman
A	0000
C	0001
B	001
F	010
H	011
D	100
G	101
E	11

### 2.2 [3]

Mensagem= A,AB,C,D,DDB,CDA,BB,ABBABA,C,D,AD,ABACDC,DB,ACDE,EEB,ACE  
(as vírgulas separam a contribuição de cada tripla)

(0,0,A) (0,1,B) (0,0,C) (0,0,D) (0,2,B) (4,2,A) (3,1,B) (2,5,A) (0,0,C) (0,0,D) (2,1,D) (6,5,C)  
(1,1,B) (5,3,E) (0,2,B) (6,2,E)

$$l_{in} = 41 \times 8 = 328 \text{ bits}$$

$$l_{out} = 16 \times (3 + 3 + 8) = 224 \text{ bits}$$

$$\text{Taxa de compressão} = \frac{328-224}{328} = 31.7\%$$

### 2.3 [2]

Mensagem= B,C,BA,C,BAC,BACA,CBACC,CCCCCA,AAAAB,BC (as vírgulas separam a contribuição de cada tripla)

## Grupo III [6 valores]

### 3.1 [2]

Exemplo de pseudocódigo (com ou sem especificação vértice inicial):

```
DepthFirstSearch(v0)
  para todos os vertices v
    num(v)=0;
  i=1;
  DFS(v0);
  enquanto existe vertice v com num(v)=0
    DFS(v)
end
```

```

DFS (v)
  num(v)=i;
  i=i+1;
  para todos os vertices u adjacentes a v
    Se num(u)=0
      DFS(u);
end

```

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

#### 4.1 [0,5]

Grafo simples, não orientado, ponderado.

#### 4.2 [0,5]

Tabela de adjacências	
Vértice	Vértices Adjacentes
a	b, c, d, e
b	a, e
c	a, f
d	a, g, h
e	a, b, f, g
f	c, e, h, i
g	d, e, h
h	d, f, g, i
i	f, h

#### 4.2 [3]

Algoritmo de Dijkstra									
Iteração	Inicialização	1	2	3	4	5	6	7	8
Vértice Ativo		a	b	d	e	c	g	h	f
a	0								
b	$\infty$	1							
c	$\infty$	2	2	2	2				
d	$\infty$	1	1						
e	$\infty$	1	1	1					
f	$\infty$	$\infty$	$\infty$	$\infty$	4	4	4	3	
g	$\infty$	$\infty$	$\infty$	2	2	2			
h	$\infty$	$\infty$	$\infty$	2	2	2	2		
i	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	5	4

A distância mais curta entre o vértice a e o vértice i é 4. O caminho é a, d, h, f, i.

FIM