



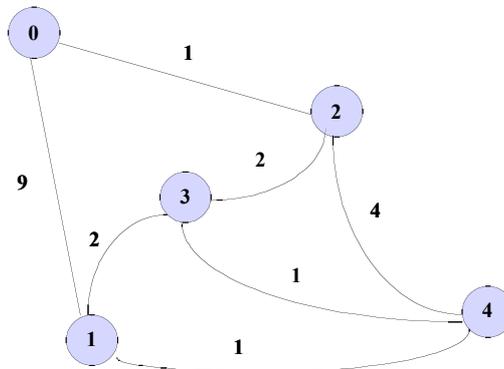
**NOTAS:**

Leia com atenção cada questão antes de responder. A interpretação do enunciado de cada pergunta é um factor de avaliação do teste. Pode utilizar elementos pessoais de consulta. A duração do teste é 2h00. O enunciado contém 7 páginas que devem ser entregues com a resposta ao teste.

NOME: \_\_\_\_\_ Nº Aluno: \_\_\_\_\_

**Questão 1**

Considere o contexto do trabalho prático nº 3 e suponha que lhe é dado o seguinte diagrama que representa uma rede com 5 nós. De acordo com o algoritmo de encaminhamento Bellman-Ford, baseado na troca de vectores de distância, resolva as questões que a seguir se colocam.



Pretende-se desenvolver uma função `rtInit()` genérica que inicializa as tabelas de encaminhamento de qualquer nó  $i$ , com  $i=\{0,1,2,3,4\}$ , segundo a seguinte especificação:

```
void rtInit( int nid, int *cCosts, RTable *rt)
```

Esta função deve receber a identificação de um nó (na variável `nid`), um vector de custos (`cCosts`) que contém em cada posição `cCosts[i]` os custos para os restantes nós. A função deve devolver a tabela de encaminhamento inicial do nó identificado por `nid`. Antes de retornar, a função `rtInit()` deve ainda enviar anúncios de actualização. A tabela de encaminhamento (`RTable`) é definida do seguinte modo:

```
typedef struct {  
    int nextHop[5];  
    int minCost[5];  
} RTable;
```



**c)** Considere, por hipótese, que num dado instante da evolução do cálculo da tabela de encaminhamento no nó 1, se chegava à seguinte situação:

<b>Tabela: Nó 1</b>	Destino: 0	Destino: 1	Destino: 2	Destino: 3	Destino: 4
Via:	0	1	4	4	4
Custo	9	0	5	2	1

Supondo que chegava ao nó 1 um anúncio do nó 3 com a seguinte informação de custos:

<b>Anúncio: do Nó 3</b>	Destino:0	Destino: 1	Destino: 2	Destino: 3	Destino: 4
<b>Custo</b>	999	2	2	0	1

Após a recepção do anterior anúncio, como ficaria a nova tabela de encaminhamento do nó 1 ? (preencha na tabela seguinte)

<b>Tabela: Nó 1</b>	Destino	Destino	Destino	Destino	Destino
Via:					
Custo					

**d)** Considere a nova tabela de encaminhamento calculada em **c)**, suponha que a seguir o nó 4 “aprenderia”, por um anúncio vindo do nó 1, que é possível chegar ao nó 0 via o nó 1 com custo 10. Esta situação, por hipótese, provocaria a alteração da tabela de encaminhamento no nó 4. Face a essa alteração, o nó 4 estaria pronto a anunciar aos seus vizinhos os novos custos segundo um anúncio da seguinte forma

<b>Anúncio: do Nó 4</b>	Destino:0	Destino: 1	Destino: 2	Destino: 3	Destino: 4
<b>Custo</b>	10	1	4	1	0

Nas condições colocadas por hipótese, será que este anúncio deve ser enviado, nesta forma exacta, a todos os nós vizinhos do nó 4?

<p>Sim ou Não: _____</p> <p>Se sim, indique qual ou quais e justifique adequadamente a resposta.</p>
--

## Questão 2

No anexo I apresenta-se código C correspondente a um excerto de uma implementação do trabalho prático. Nesta implementação apresenta-se uma solução para as funções `rtUpdate()` e `rtLinkHandler()`, definidas de forma genérica. Adicionalmente define-se a função auxiliar `rtAnnounce` que deve ser usada para anunciar os novos custos ao nós vizinhos.

Complete o código apresentado, preenchendo as caixas, de forma que seja implementada a técnica de *"split horizon with poisoned reverse"*.

## Questão 4

No âmbito do trabalho prático realizado, comente as seguintes afirmações, indicando se as mesmas são verdadeiras ou falsas. **Justifique as afirmações.**

- a) Cada nó,  $i$ , mantém um vector de custos mínimos, em que a posição  $j$  indica uma estimativa do custo mínimo do encaminhamento entre o nó  $i$  e o nó  $j$ . Em consequência da recepção no nó  $i$  de um anúncio dos custos mínimos relativos ao nó  $k$ , é impossível que o valor da posição  $j$  ( $j < k$ ) dos custos mínimos, em  $i$ , seja substituída por um valor superior ao actualmente presente.

Verdadeiro  porque... Falso  porque...

- b) Num nó  $i$ , a matriz que mantém a distância do nó  $i$  a cada um dos outros nós através de cada um dos nós com ligações directas a  $i$  pode ser usada para melhorar o funcionamento do sistema quando existe uma alteração no custo associado a uma ligação da rede.

Verdadeiro  porque... Falso  porque...

## Questão 5

No algoritmo de Bellman-Ford, cada router anuncia aos seus vizinhos directos os destinos que alcança e o custo com que os alcança. É possível modificar este algoritmo para que cada um destes anúncios passe a incluir também o caminho que o router utiliza para alcançar cada um dos destinos que conhece. Por exemplo, o router  $R$  anuncia aos seus vizinhos uma lista  $d[1], d[2], \dots, d[n]$  de destinos e para cada um deles ( $d[i]$ ) indica não só o custo com que o alcança ( $c$ ) mas também a lista de routers ( $R_1, R_2, \dots, R_k$ ) intermédios que são usados para o alcançar. Um anúncio de um destino  $d[i]$  tem então a forma:  $d[i], c, R[i,1], R[i,2], \dots, R[i,k]$ .

Inicialmente um router apenas consegue anunciar os caminhos para os seus vizinhos directos e o caminho terá comprimento 1. Mas à medida que vai recebendo anúncios, não só enriquece a sua tabela de encaminhamento (routing) com mais destinos, como também anota o caminho (por hipótese óptimo) que utiliza para os alcançar.

- a) Indique qual a nova estrutura usada para implementar a tabela de encaminhamento supondo que queria introduzir esta variante do algoritmo no seu trabalho, no contexto do enunciado da questão 1.

- b) Como seria então a estrutura de uma anúncio no mesmo contexto ?

- c) Em termos de complexidade espacial compare a tabela de encaminhamento e os anúncios na "antiga" e na nova versão do algoritmo.

- d) À parte a nova estrutura da tabela de encaminhamento e dos anúncios, o algoritmo pode continuar a ser, no essencial, igual à versão que implementou, isto é, usar a distância a cada vizinho e os custos presentes nos seus anúncios para tomar decisões sobre quais os melhores caminhos. No entanto, o conhecimento dos caminhos completos até ao destino, permite lidar de uma forma mais correcta com um problema presente no algoritmo de Bellman-Ford tal como o implementou nas práticas ou estudou nas aulas teóricas. Explique qual o problema e como é que este novo algoritmo permite lidar com esse problema.

- e) Porque não é muito popular a utilização desta variante do algoritmo?

## ANEXO I

```
// Envia anúncios aos nós vizinhos. Parâmetros:
// node: nó em que a função executa
// cCosts: vector de custos directos
// rt: tabela de encaminhamento no nó
void rtAnnounce( int node, int *cCosts, RTable *rt) {
```

```
}
// Actualiza tabela de encaminhamento aquando da recepção de um anúncio.
// Parâmetros:
// u: anúncio recebido
// cCosts: vector de custos directos
// rt: tabela de encaminhamento no nó
void rtUpdate(struct rtpkt *u, int *cCosts, RTable *rt) {
    int i, changed = 0;
    for( i = 0 ; i < 5 ; i++ ) {
        if( i == u->destid) continue;
        int newCost = u->mincost[i]+cCosts[u->sourceid];

        if( newCost < rt->minCost[i] ) {
            rt->minCost[i] = newCost ;
            rt->nextHop[i] = u->sourceid;
            changed = 1;
        }
        if( newCost > rt->minCost[i] && [ ] ) {
            if( newCost < cCosts[i] ) {
                rt->minCost[i] = newCost ;
                rt->nextHop[i] = [ ] ;
                changed = 1;
            } else {
                rt->minCost[i] = cCosts[i];
                rt->nextHop[i] = [ ] ;
                changed = 1;
            }
        }
    }
    if( changed)
        rtAnnounce(u->destid, cCosts, rt);
}
```

```
// Função que processa a modificação do custo de um link entre dois nós
// Parâmetros:
// linkid: nó destino do link cujo custo foi modificado
// newCost: novo custo do link
// cCosts: vector de custos directos
// rt: tabela de encaminhamento no nó
void rtLinkHandler(int linkid, int newCost, int *cCosts, RTable *rt) {
```

```
}
```