



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.

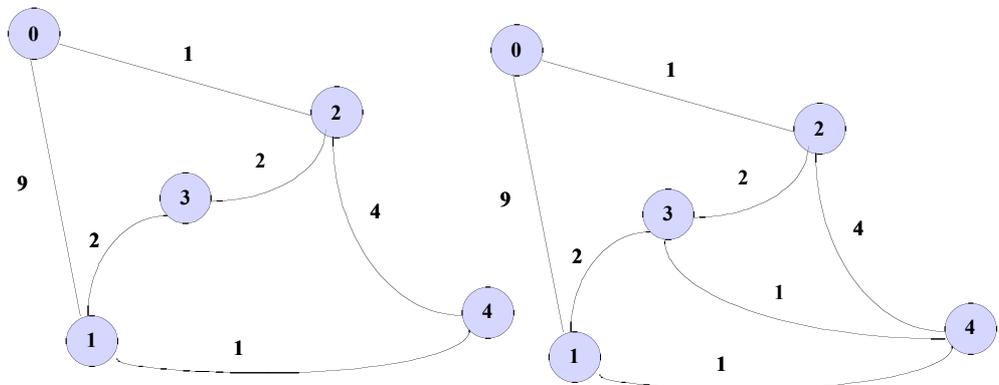
A duração do teste é 2h00.

O enunciado contém 7 páginas que devem ser entregues com a resposta ao teste.

NOME: _____ N° 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 rdInit() genérica que inicializa a tabela de distâncias de qualquer nó i , com $i=\{0,1,2,3,4\}$, segundo a seguinte especificação:

```
void rdInit( int nid, int *cCosts, RDistance *rd)
```

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 inicial de distâncias do nó identificado por nid. Antes de retornar, a função rdInit() deve ainda enviar os anúncios de actualização necessários. A tabela de encaminhamento (RDistance) é definida do seguinte modo:

```
typedef struct {  
    int distance[5][5];    //distance[i][j] : estimative da distância ao nó I sendo
```


b) Diga com que argumentos de entrada se invoca a função `rtInit()` no caso dos nós 3 e 4 e quais as tabelas de encaminhamento inicialmente calculadas.

Argumentos de entrada no Nó 3:

Argumentos de entrada no Nó 4:

Tabela: Nó 3	Destino: 0	Destino: 1	Destino: 2	Destino: 3	Destino: 4
Via:					
Custo					

Tabela: Nó 4	Destino: 0	Destino: 1	Destino: 2	Destino: 3	Destino: 4
Via:					
Custo					

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

Tabela: Nó 1	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:

Anúncio: do Nó 3	Destino:0	Destino: 1	Destino: 2	Destino: 3	Destino: 4
Custo	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)

Tabela: Nó 1	Destino	Destino	Destino	Destino	Destino
Via:					
Custo					

d) Após a nova tabela de encaminhamento ter sido calculada em **c)**, mais cedo ou mais tarde, 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 provoca 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

Anúncio: do Nó 4	Destino:0	Destino: 1	Destino: 2	Destino: 3	Destino: 4
Custo	10	1	4	1	0

Será que este anúncio deve ser enviado, nesta forma exacta, a todos os nós vizinhos do nó 4?

Sim ou Não: _____
 Se sim, indique qual ou quais e justifique adequadamente a resposta.

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 `rdUpdate()` e `rdLinkHandler()`, definidas de forma genérica. Adicionalmente define-se a função auxiliar `rdAnnounce` 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 3

Suponha que a tabela de distâncias

Ao mudar o custo de um link, adicionar diferença do link é suficiente para que a coluna da tabela de distâncias relativa ao nó em questão fique imediatamente correcta???

Questão 4

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

- 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 o alcança cada um deles. 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 alcançar o destino d_i . Um anúncio 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 anotará 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.

- b) Como seria então a estrutura de uma anúncio?

- 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 do caminho completo até ao destino que vem nos anúncios, e o que está presente nas tabelas de encaminhamento, permite lidar de uma forma

definitivamente correcta com um problema presente no algoritmo de Bellman-Ford tal como o implementou ou estudou nas aulas teóricas. Explique qual o problema e como é que este novo algoritmo permite lidar com esse problema.

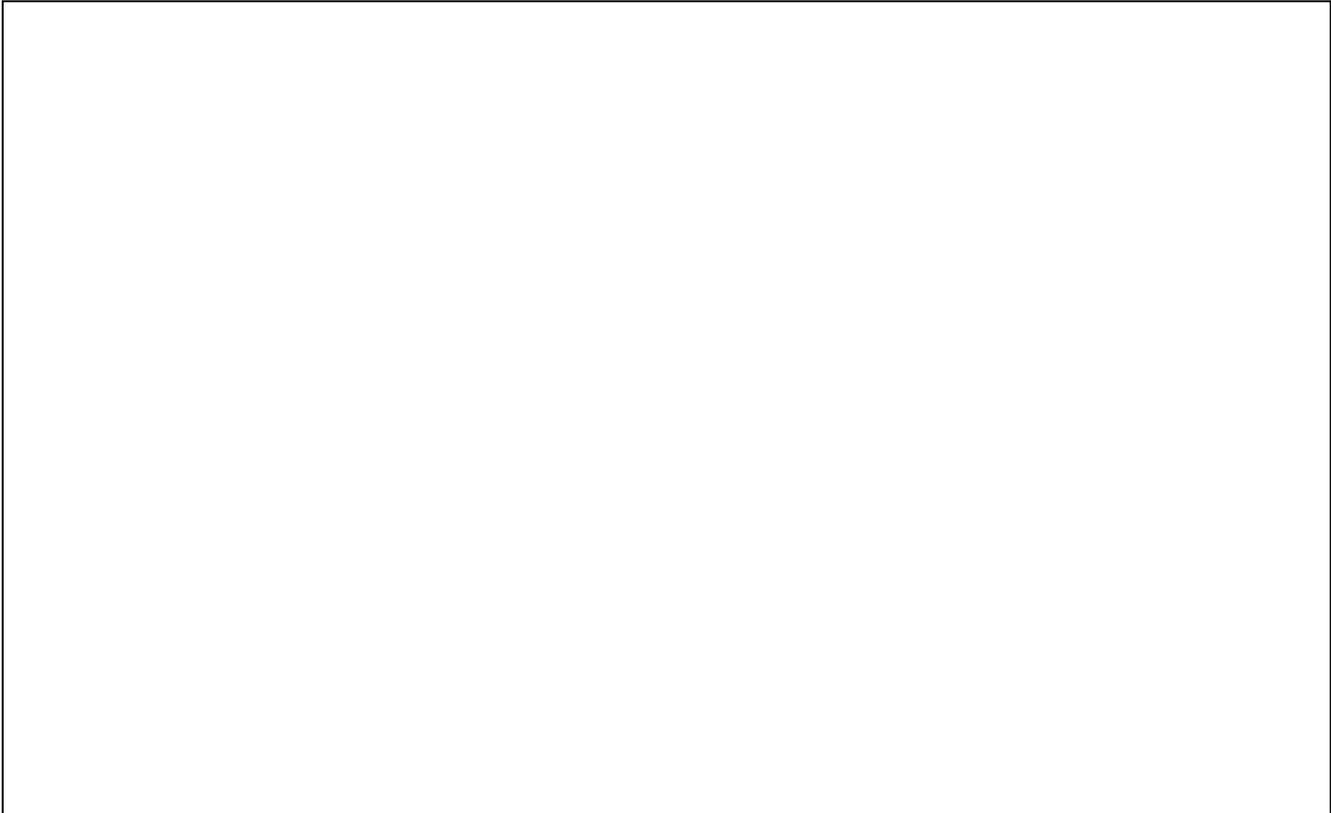
e) Porque não é muito popular a utilização desta variante do algoritmo?

ANEXO I

```
// Envia anúncios aos nós vizinhos. Parâmetros:  
// node: nó em que a função executa  
// cCosts: vector de custos directos  
// rd: tabela de distâncias no nó  
void rdAnnounce( int node, int *cCosts, RDistance *rd) {
```

```
}
```

```
// Atualiza tabela de encaminhamento aquando da recepção de um anúncio.  
// Parâmetros:  
// u: anúncio recebido  
// cCosts: vector de custos directos  
// rd: tabela de distâncias no nó  
void rdUpdate(struct rtpkt *u, int *cCosts, RDistance *rd) {  
    int i, changed = 0;  
    for( i = 0 ; i < 5 ; i++ ) {
```



```
    }  
    if( changed)  
        rdAnnounce(u->destid, cCosts, rd);  
}
```

```
// 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
// rd: tabela de distâncias no nó
void rdLinkHandler(int linkid, int newCost, int *cCosts, RDistance *rd) {
```

```
}
```