Exame de Recurso — Redes de Computadores de 2018/2019

(Preencha sem falta todos os campos abaixo)

Nome: _____ **N.º** _____

Sala do exame: _____ **Edifício VII** **Versão:** **A** **Venho repetir o teste 2 (S/N):** _____

Duração: 2 horas e 15 minutos acrescidos de 15 minutos de tolerância
A interpretação das questões é da sua responsabilidade.
Não pode usar dispositivos electrónicos de qualquer tipo.
Nas respostas de escolha múltipla com várias hipóteses, as hipóteses erradas descontam. Se seleccionar todas as hipóteses, a cotação final é nula.

RESPOSTA ÀS QUESTÕES - PODE USAR LÁPIS

1.1) 4
outros diminuir 25%

1.2) 2

1.3) 2

1.4) 4

1.5) 3,4,5 cada vale 33,3%
descontar 15% por errada

1.6) 1,7,8 idem anterior

(medianas)

1.7 a) 4 **1.7 b)** 4 **1.7 c)** 1 **1.7 d)** 20 **(medianas)**

1.8 a) 20

1.8 b) 7 ou 8

(difícil)

2.1) 4 **2.2)** 3 **2.3)** 1 **2.4)** 2 **(mais do mesmo)**

2.5) nenhuma!

2.6) 1,2 cada vale 50%
outras, descontar 50%

(2.5 difícil)
(2.6 mediana)

AS PERGUNTA 2.7 e 2.8 DEVEM SER RESPONDIDAS NO RESPECTIVO ENUNCIADO NO VERSO

2.7)

2.8)

(2.7 difícil)
(2.8 mediana)

2.7 - VERSÃO A - vale 2 valores) Considere a rede da pergunta **2.6** com as seguintes hipóteses: o tempo de propagação do canal que liga R3 a R5 e vice-versa é 10 vezes **maior** que o dos outros canais; a rede usa um protocolo de encaminhamento baseado no algoritmo Bellman-Ford com anúncios periódicos, **com memorização dos anúncios e *split horizon with poison reverse***.

É possível entrar-se num ciclo de contagem para o infinito nesta rede? Justifique a sua resposta.

SIM podem (acertar vale 20%)

Se o canal que liga R1 a R2 for abaixo, todos os routers recebem o aviso de que a distância agora é infinito.
Mas R5 e R4 conhecem um caminho pior (em backup) via o outro pois como não o usam, é anunciado ao outro.
Como o tempo de propagação do link R5/R4 é muito grande, R4 e R5 não têm tempo de se aperceberem que esses caminhos também já não estão disponíveis, e podem reintroduzi-los na rede o que conduz ao ciclo.

a justificação vale 80%

2.8 - VERSÃO A - vale 2 valores) Numa rede, os comutadores têm vários vizinhos, a cada um dos quais estão ligados directamente por um canal distinto ponto a ponto. Por hipótese, os **N** destinos existentes na rede são os comutadores, que são identificados por uma número inteiro de **1 a N**. Na rede usa-se o algoritmo Bellman-Ford na versão ***split horizon with poison reverse*** para actualizar as tabelas de encaminhamento.

Um anúncio é um vetor com **N** entradas em que cada entrada indica o **custo** para chegar ao destino correspondente à mesma. Por exemplo, **costs (i)**, indica o custo com que o emissor deste anúncio chega ao destino **i.**

Um comutador tem uma tabela de encaminhamento que suporta vários métodos, entre os quais:

int getNextHop (destination) - devolve o vizinho que encaminha para **destination** ou null se a tabela de encaminhamento desconhece destination

int getCost (destination) - devolve o custo do caminho deste comutador até **destination** ou **infinity** se se a tabela de encaminhamento desconhece **destination**

update (destination, distance, gateway) - cria / atualiza a entrada da tabela referente a **destination**.

Escreva o pseudo código usado pelo comutador para gerar o anúncio que vai enviar ao vizinho **K.**

```
RoutingTable rt = new RoutingTable(N); // a tabela de encaminhamento do comutador contendo N entradas
int[] costs = new int[N];  // o anúncio a gerar

 for ( int i = 1; i <= N; i++ ) {                                  // ciclo vale 10%

      if ( i == mySelf ) cost[i]=0 ;  // se esquecer não desconta nada rt.getCost(mySelf) deve ser 0

      else if ( rt.getNextHop(i) == K ) costs[i] = infinity ;      //   45%

      else costs[i] = rt.getCost(i);                               //   45%

}
```