Universidade do Porto — L:EIC

Desenho de Algoritmos — 2021/22

Exame (30.06.2022) - Parte Prática — *90 minutos*

N.º ____________ Nome ______________________________

Atenção: Não serão prestados esclarecimentos durante a prova. Leia pf as perguntas com atenção.
Deve responder em **quatro folhas**:

- 1ª Folha: questões **1a)** e **1b)**;
- 2ª Folha: questões **1c)** e **1d)**;
- 3ª Folha: questão **2**;
- 4ª Folha: questão **3**.

1. Uma empresa dispõe de N caixas de cerejas para distribuir por M lojas. As caixas têm todas o mesmo peso e o conteúdo não pode ser repartido. Devido a características específicas de cada loja, os valores que as lojas estão dispostas a pagar variam não só de loja para loja, como também consoante o número de caixas que recebem. Seja v_{jk}, com $v_{jk} \in \mathbb{Z}_0^+$, o valor da oferta da loja j por k caixas. Para cada loja j, os valores v_{jk} crescem com k, até ao limite máximo pretendido pela loja, não sendo necessariamente proporcionais a k. A empresa pretende distribuir as N caixas, se possível, e maximizar o valor obtido. Não é necessário enviar caixas para todas as lojas. Denote-se por $L(k, j)$ o **valor máximo** se se distribuir k caixas pelas j primeiras lojas, para $1 \leq j \leq M$ e $0 \leq k \leq N$, e por $S(k, j)$ **uma solução ótima** correspondente, definida por um conjunto de pares (p, x_p), em que p identifica a loja e $x_p > 0$ a quantidade enviada.

a) $\boxed{1.0 \text{ v}}$ Por análise da instância seguinte, explique porque é que a estratégia *"enviar o máximo possível à loja que oferecer mais (em caso de empate, optar por uma qualquer dessas) e, se sobrar alguma caixa, proceder de forma idêntica para as caixas e lojas sobrantes"* não resolve corretamente o problema.

Exemplo: $N = 4$ e $M = 3$.

		Caixas			
		1	**2**	**3**	**4**
Lojas	**1**	2	4	5	6
	2	4	5	7	0
	3	1	6	9	14

$L(1,2) = 4$	$S(1,2) = \{(2,1)\}$
$L(2,2) = 6$	$S(2,2) = \{(1,1),(2,1)\}$
$L(3,2) = 8$	$S(3,2) = \{(1,2),(2,1)\}$
$L(3,3) = 10$	$S(3,3) = \{(2,1),(3,2)\}$
$L(4,3) = 14$	$S(4,3) = \{(3,4)\}$

b) $\boxed{2.0 \text{ v}}$ **Usando C/C++ ou Java**, escreva uma função que recebe N, M e a matriz V, de dimensão $M \times N$, como parâmetros, e retorna os valores $L(N, M)$ e $S(N, M)$ que seriam obtidos pela **estratégia ganonciosa** definida na alínea **1a)**. É discutível se V deveria ser suportado por outra estrutura de dados (que não uma matriz), mas considere essa representação para V na implementação.

c) $\boxed{1.5 \text{ v}}$ Usando **pseudocódigo** (ou C/C++ ou Java), escreva um algoritmo que determine valores corretos para $L(N, M)$ e $S(N, M)$, por aplicação de **programação dinâmica**, assumindo que $v_{jN} > 0$, para todo j. Deverá ter **complexidade temporal** $O(MN^2)$. Comece por definir matematicamente $L(k, j)$, por uma recorrência (i.e., definição recursiva), para $1 \leq j \leq M$ e $0 \leq k \leq N$.

d) $\boxed{1.0 \text{ v}}$ Assumindo que P $\neq$ NP, diga se, dada uma instância (N, M, V), com $v_{jN} > 0$, para todo j, e dado um inteiro T, o problema de **decidir se é possível obter pelo menos o montante** T é NP-completo. Justifique a resposta sucintamente.

(Continua)

N.º ______ Nome ______

2. A empresa já sabe quantas caixas de cerejas deve enviar a cada loja, isto é, dispõe da solução $S(N, M)$. Irá subcontratar o serviço de entregas a **duas** empresas, devendo minimizar a diferença (em valor absoluto) entre o número total de caixas que cada uma entregará. Cada loja recebe entregas de uma delas, no máximo.

a) $\boxed{\text{2.0 v}}$ Formule matematicamente o problema, com variáveis de decisão boolenas. Indique quais são os dados e a interpretação das variáveis de decisão, das restrições e da função objetivo. Note que, na formulação podemos (e devemos) fixar qual das duas empresas entregaria o maior número de caixas, se necessário.

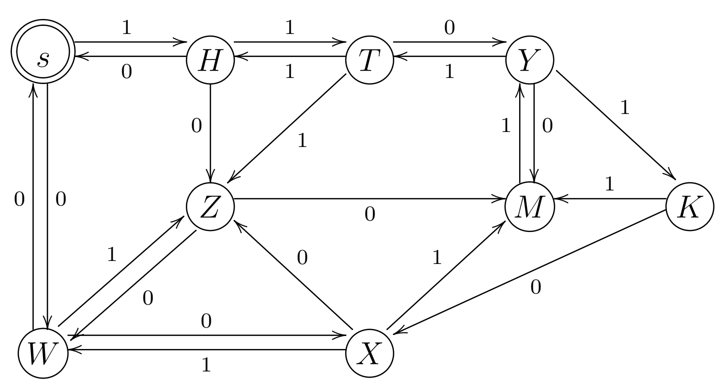
b) $\boxed{\text{1.5 v}}$ Apresente um **algoritmo polinomial** que, dados M e $S(N, M)$, obtém uma solução para o problema, possivelmente não ótima, por aplicação de uma **estratégia gananciosa** (*greedy*). A descrição deve ser concisa e clara (mas não é necessário uma codificação detalhada). Indique que **complexidade temporal** teria, explicando sucintamente.

c) $\boxed{\text{0.5 v}}$ Assumindo que P $\neq$ NP, explique porque é que o problema de otimização enunciado não poderia ser resolvido em tempo polinomial (ainda que o possa ser para algumas instâncias).

3. Devido ao aumento de combustíveis, uma das empresas de entregas irá efetuar a distribuição com recurso a estafetas que se deslocam em bicicletas com atrelados. Cada estafeta pode efetuar **várias viagens** e, em cada instante, **todas as caixas que transporta são para a mesma loja**. Havendo congestionamentos na rede urbana, a empresa pretende determinar qual seria **o melhor caminho** desde a empresa até cada uma das lojas (para todas as lojas). Pretende caminhos **sem congestionamentos ou com no máximo um ramo congestionado**. Cada um desses caminhos deve ter o **menor número de ramos** possível. Se o caminho mais curto tiver algum congestionamento, deve também obter o melhor caminho sem congestionamentos, se existir. Em caso de empate, pode ser indicado um qualquer. Admita que a rede urbana é dada por um grafo dirigido $G = (V, E, p)$, em que $p(e) \in \{1, 0\}$ indica se o ramo $e \in E$ está ou não congestionado, sendo 1 se estiver. A empresa é identificada por um nó s e as lojas pelos outros nós.

Para a caraterização de um tal caminho de s até v, para $v \in V \setminus \{s\}$, deve ser indicado um **terno**: se tem ou não congestionamento, qual é o seu comprimento e qual é o nó que antecede v nesse caminho.

a) $\boxed{\text{1.0 v}}$ Por aplicação de uma estratégia baseada em pesquisa em largura (BFS), mas **em que cada nó pode entrar na fila mais do que uma vez** (mas nunca mais do que duas vezes), resolva a instância seguinte. Na resolução, deve indicar que nó sai da fila e que nós entram para a fila em cada passo, para permitir compreender o algoritmo que está a aplicar, o qual **deverá ter complexidade temporal** $O(|E| + |V|)$.



b) $\boxed{\text{1.5 v}}$ Traduza (em **pseudocódigo**) o algoritmo que concebeu para resolver o problema em tempo $O(|E|+|V|)$, sendo G e s dados. Recorde que, para cada $v \in V$, pode ter de caraterizar 0, 1 ou 2 caminhos, por **ternos**, como se descreveu acima. Serão dois caminhos se não forem comparáveis (i.e., se em termos de distância ou do número de congestionamentos forem ambos ótimos).

(Fim)