Universidade do Porto — L:EIC

Desenho de Algoritmos — 2021/22

Exame (30.06.2022) - Parte Teórica - A — *30 minutos*

N.º [ ] Nome [ ]

Atenção: Não serão prestados esclarecimentos durante a prova. Leia pf as perguntas com atenção. As respostas devem ser assinaladas na tabela seguinte, usando **LETRAS MAIÚSCULAS**, sem rasuras. Cotação: 8 valores. Três respostas erradas, descontam uma certa. Ausência de resposta não desconta.

Respostas:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

1. Seja $\mathcal{T}$ um conjunto de n tarefas. A tarefa t_j tem de decorrer no intervalo $[s_j, f_j]$, ou seja, começar no instante s_j e terminar em f_j, para $1 \leq j \leq n$. Em cada instante só uma tarefa pode estar a decorrer, mas podemos iniciar outra tarefa no instante exato em que termina. Pretendemos **maximizar o número de tarefas realizadas**. Para tal, ordenaremos as tarefas segundo um certo critério e, seguindo essa ordem, escolhemos as que não entram em conflito com alguma anteriormente selecionada. Qual das ordens seguintes produz sempre uma solução ótima?

a) Ordem crescente de duração $(f_j - s_j)$.
b) Ordem crescente de instante inicial (s_j).
c) Ordem crescente de instante final (f_j).
d) Nenhuma das ordens indicadas.

2. Considere o *problema do labirinto*, num espaço bidirecional de células cujos estados podem representar a presença de um obstáculo ou uma passagem livre. Qual seria a melhor abordagem para conceber um algoritmo para procurar um caminho desde uma origem dada até uma possível saída do labirinto?

a) Programação dinâmica **b)** Gananciosa **c)** Pesquisa com retrocesso **d)** Divisão e conquista

3. A sucessão de Fibonacci é definida por $f_0 = 0$, $f_1 = 1$ e $f_n = f_{n-1} + f_{n-2}$, para $n \geq 2$. Considerando as duas funções, sem atender a erros de *overflow*, é verdade que:

```
int fib(int n) {
   if (n <= 1) return n;
   return fib(n-1)+fib(n-2);
}
int fib_B(int n) {
   if (n <= 1) return n;
   int a = 0, b = 1, c, i;
   for(i = 2; i <= n; i++)
    { c = a + b; a = b; b = c; }
   return b;
}
```

a) Ambas efetuam $O(n)$ adições, para $n \geq 2$.

b) `fib_B` explora programação dinâmica.

c) `fib` baseia-se em divisão-e-conquista.

d) `fib` é mais eficiente do que `fib_B`.

4. Qual das afirmações seguintes sobre grafos não dirigidos, com $n \geq 3$ nós e $m > 0$ ramos, é **incorreta**?

a) Nem todos os grafos conexos e bipartidos têm um circuito de Euler.

b) Um grafo completo tem sempre um ciclo de Hamilton.

c) Se um grafo tem um circuito de Euler então tem um caminho de Euler.

d) Existe um algoritmo com complexidade $O(n^2)$ para verificar se um grafo tem um circuito de Euler.

N.º | | Nome | |

5. A função `hasSum` (em C++) determina se um *array* `a` com `n` inteiros positivos contém um subconjunto de elementos cuja soma é um inteiro `x`. Que tipo de abordagem implementa?

```
bool hasSum(int a[], int n, int x) {
   if (x == 0) return true;
   if (n == 0 || x < 0) return false;
   return hasSum(a, n-1, x-a[n-1]) || hasSum(a, n-1, x);
}
```

a) Programação dinâmica **b)** Gananciosa **c)** Pesquisa com retrocesso **d)** Divisão e conquista

6. A Câmara de uma pequena cidade pretende definir a melhor rota, em termos de distância percorrida, para um camião de recolha de lixo, que sai da estação de tratamento de resíduos, passa por todas as ruas, e regressa à estação. Se necessário, no percurso pode passar várias vezes na mesma rua. Este problema pode ser interpretado como:

a) O problema do caixeiro viajante (TSP).

b) O problema do carteiro chinês.

c) O percurso ótimo do caixeiro viajante.

d) O caminho mínimo entre todos os pares de nós.

7. Face à matéria lecionada sobre complexidade de problemas e de algoritmos, é verdade que:

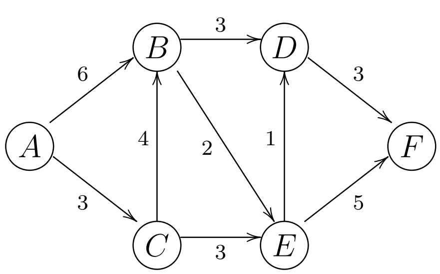
a) O problema de decidir se t é acessível de s num grafo dirigido $G = (V, E)$, com $s \neq t$, é da classe NP.

b) O problema k-SAT é NP-díficil (*NP-hard*), qualquer que seja $k \geq 2$, fixo.

c) Os problemas NP-completos não são da classe NP.

d) Todos os problemas da classe NP requerem algoritmos exponenciais.

8. O projeto representado pela rede de atividades seguinte (modelo arco-atividade) deve ser concluído o mais cedo possível. Não há partilha de recursos. É verdade que:



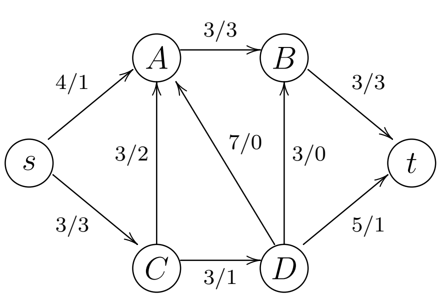
a) A duração mínima do projeto é 12.

b) As tarefas AC, CE, ED e DF são críticas.

c) A tarefa EF começará no instante 6.

d) A tarefa AB não é crítica.

9. Considere a rede de fluxo seguinte, onde c/f são pares capacidade/fluxo, e s e t são a origem e destino. É verdade que:



a) O corte $\{S, T\}$ de capacidade mínima tem capacidade 4.

b) Não é possível aumentar o fluxo porque as ligações (s, C), (A, B) e (B, t) estão saturadas.

c) O algoritmo de Edmonds-Karp encontra (s, A, C, D, t) como caminho para aumento de fluxo.

d) O valor de $f(C, D) = 1$ e de $f(D, C) = 0$.

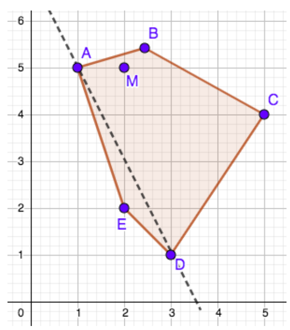
N.º | Nome

10. Para a instância seguinte do problema de emparelhamentos estáveis (*stable marriage*), com listas de preferências estritamente ordenadas, por ordem decrescente de preferência, é verdade que:

$a_1 : p_1, p_2, p_3, p_4$	$p_1 : a_4, a_2, a_3, a_1$
$a_2 : p_1, p_4, p_3, p_2$	$p_2 : a_2, a_3, a_1, a_4$
$a_3 : p_3, p_1, p_4, p_2$	$p_3 : a_1, a_2, a_4, a_3$
$a_4 : p_2, p_3, p_1, p_4$	$p_4 : a_4, a_2, a_3, a_1$

a) Não existe nenhum emparelhamento estável.
b) $\{(a_4,p_1),(a_2,p_2),(a_1,p_3),(a_4,p_4)\}$ é a solução.
c) $\{(a_1,p_3),(a_2,p_2),(a_3,p_4),(a_4,p_1)\}$ é estável.
d) $\{(a_1,p_4),(a_2,p_1),(a_3,p_3),(a_4,p_2)\}$ não é estável.

11. Considere a aplicação do método Simplex para obter uma solução ótima de um problema de programação linear no plano, cujo espaço de soluções é o polígono indicado. Os pontos assinalados têm coordenadas inteiras, exceto B. A reta AD (a tracejado) é curva de nível da função objetivo. É verdade que:



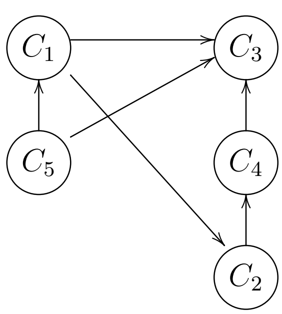
a) Se partir de D, efetua apenas uma iteração para obter a solução ótima.

b) B é a solução ótima linear que o Simplex obtém.

c) M é a solução da relaxação linear.

d) A é solução básica ótima que o Simplex obtém.

12. Seja $G = (V, E)$ um grafo dirigido, com $V = \{v_i \mid 1 \leq i \leq 20\}$, tal que G tem exatamente cinco componentes fortemente conexas, designadas por C_1, C_2, C_3, C_4 e C_5, sendo $v_1 \in C_1$, $v_2 \in C_2$, $v_3 \in C_3$, $v_4 \in C_4$ e $v_5 \in C_5$. O grafo das componentes fortemente conexas de G encontra-se abaixo. É verdade que:



a) É impossível ter $\{(v_1,v_3),(v_1,v_5)\} \subset E$.

b) No algoritmo de Kosaraju-Sharir, as componentes são reportadas pela ordem C_1, C_2, C_3, C_4, C_5.

c) No algoritmo de Kosaraju-Sharir, a primeira componente reportada é C_3.

d) Não existem $x \in C_1$ e $y \in C_2$ tais que $(x, y) \in E$.

13. O nome "Carlos" foi escrito erradamente, como "Cralis". Qual a distância de edição (Levenshtein) entre as duas grafias, com custos unitários para os três tipos de transformação?

a) 1 **b)** 2 **c)** 3 **d)** 4

14. Quantos algarismos significativos tem o número 0.05300×10^3?

a) 2 **b)** 3 **c)** 4 **d)** 5

15. Para obter um zero da função $f(x) = \cos(x) - x$, para $x \in \mathbb{R}^+$, isto é, obter uma aproximação de uma solução da equação $\cos(x) = x$, com erro absoluto máximo de 1×10^{-5}, seria melhor aplicar:

a) O algoritmo de Gauss-Jordan.

b) O método Simplex partindo de $x = 0$.

c) Bisseções sucessivas, partindo do intervalo $[0, 3]$.

d) O algoritmo de Floyd-Warshall.

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 gananciosa** 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)

Universidade do Porto - L:EIC - Desenho de Algoritmos – Exame 30.06.2022

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) $\boxed{2.0 \text{ v}}$ 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.

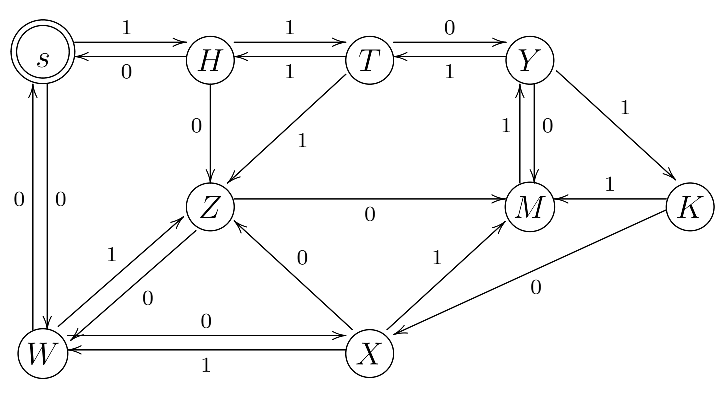
b) $\boxed{1.5 \text{ v}}$ Apresente um **algoritmo polinomial** que, dados M e $S(N, M)$, obtém uma solução para o problema, possivelmente não ótima, por aplicação de uma **estratégia gananciosa** (*greedy*). A descrição deve ser concisa e clara (mas não é necessário uma codificação detalhada). Indique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

c) $\boxed{0.5 \text{ v}}$ 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) $\boxed{1.0 \text{ v}}$ 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) $\boxed{1.5 \text{ v}}$ 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)**