INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos

Ano lectivo 2018/2019 Repescagem 2º Teste - versão A

RESOLUÇÃO DA REPESCAGEM 2º TESTE

I. (2,0 + 4,0 + 2,0 = 8,0 val.)

I.a)

	Valor	Lista de índices
Fracionário	5,75	2, 5, 3
PD	5	2, 5
Greedy	5	2, 5

I.b)

Primeira	Z =-48	x_1 =6	x_2 =0	x_3 =0
Ótima	Z =-6	x_1 =27	x_2 =14	x_3 =0
Dual Ótima	Z' =-6		$y_1 = 1$	$y_2 = 6$

I.c)

	a	b	c	d	e	f
Codificação	11	011	00	10	0100	0101

Total Bits	148

II. (2,0 + 2,0 = 4,0 val.)

II.a)

`m[1,3]`	`m[2,4]`	`m[1,4]`	Parênteses
10	10	14	$(A_1 \times A_2) \times (A_3 \times A_4)$

II.b) $\langle$ XXX $\rangle$

III. (2,0 + 2,0 = 4,0 val.)

III.a)

$\delta(4,b)$	$\delta(5,a)$	$\delta(6,b)$	$\delta(7,a)$	$\delta(7,b)$
5	4	1	0	3

III.b)

$P =$	a	a	b	a	a	a	b	a	a	b
i	1	2	3	4	5	6	7	8	9	10
$\pi[i]$	0	1	0	1	2	2	3	4	5	3

IV. (2,0 + 2,0 = 4,0 val.)

IV.a)

	a)	b)	c)	d)	e)
Resposta	V	D	V	V	V

IV.b) $\langle$ XXX $\rangle$

I. (2,0 + 4,0 + 2,0 = 8,0 val.)

I.a) Considere uma instância do problema da mochila com 5 objectos. O peso máximo que a mochila pode transportar é $M = 10$. O peso do i-ésimo objeto é denotado por p_i e o valor do i-ésimo objeto é v_i. Estes valores são apresentados na seguinte tabela:

i	1	2	3	4	5
v_i	2	2	3	1	3
p_i	7	3	8	5	5

Calcule os valores máximos obtidos para os seguintes problemas:

- O problema fracionário.
- O problema não fracionário, utilizando um algoritmo baseado em programação dinâmica.
- O problema não fracionário, utilizando um algoritmo *greedy*.

Indique a lista dos objectos selecionados, pelo respectivo índice. Na solução do problema fracionário o objecto que é partido deve ser o último da lista.

I.b) Considere o seguinte programa linear:

$$\begin{array}{lrcr} \text{Maximizar} & -8x_1+15x_2-8x_3 & & \\ \text{Sujeito a} & -2x_1+3\ x_2+6x_3 & \leq & -12 \\ & -\ x_1+2\ x_2+3x_3 & \leq & 1 \\ & x_1,x_2,x_3 & \geq & 0 \end{array}$$

Indique o valor da função objectivo e o respectivo valor das variáveis x_1, x_2 e x_3 na **primeira solução exequível** encontrada pelo algoritmo Simplex. Em caso de empate em algum critério de aplicação do algoritmo, aplique a regra de Bland. Ou seja, escolha a variável de menor índice.

Indique também o valor da função objectivo e o respectivo valor das variáveis para a solução ótima e para a solução do sistema dual, com a variável y_1 associada à primeira restrição e a variável y_2 associada à segunda restrição.

I.c) Considere o problema de compressão de dados de um ficheiro usando a codificação de Huffman. Indique o código livre de prefixo ótimo para cada caractere num ficheiro com 66 caracteres com o seguinte número de ocorrências: $f(a) = 22, f(b) = 10, f(c) = 11, f(d) = 20, f(e) = 1, f(f) = 2$. Quando constrói a árvore, considere o bit 0 para o nó com menor frequência.

Indique também o total de bits no ficheiro codificado.

II. (2,0 + 2,0 = 4,0 val.)

II.a) Considere o problema de multiplicar cadeias de matrizes. O objetivo é determinar por que ordem devem ser feitas as multiplicações por forma a minimizar o número total de operações escalares que precisam ser efetuadas.

Considere uma sequência com 4 matrizes $A_1(2\times 3), A_2(3\times 1), A_3(1\times 2), A_4(2\times 2)$, com as respetivas dimensões entre parênteses. Resolva este problema preenchendo a matriz `m[i,j]` que guarda o menor número de multiplicações que precisam de ser executadas para obter o produto das matrizes de A_i a A_j. Indique os valores de `m[1,3]`, `m[2,4]` e `m[1,4]`. Indique também a colocação de parênteses que obtém o valor indicado em `m[1,4]`.

II.b) Nesta pergunta iremos determinar se uma string resulta de intermisturar outras duas. Dadas duas strings, as suas letras são misturadas para obter uma string final, usando um processo que preserva a sequência das letras nas strings originais. A mistura consiste apenas em qual das strings originais é que é selecionada em cada instante, sendo que todas as letras das strings originais devem ser utilizadas. Por exemplo, dadas as strings $S = ABC$ e $R = DEF$, podemos obter a string $ABDCEF$. Também podemos obter a string $DEFABC$. Contudo, a string $ADBC$ não seria um resultado válido visto que não utiliza as letras E e F de R. A string $ABCDFE$ também não seria um resultado válido, visto que a ordem das letras E e F não corresponde à ordem original em R.

Pretendemos determinar, utilizando programação dinâmica, se uma string T pode ser obtida como a mistura das strings S e R.

Definimos uma tabela `M[i, j]` em que $0 \leq i \leq |S|$ e $0 \leq j \leq |R|$, que guarda 0 ou 1. Quando o prefixo das primeiras i letras de S pode ser misturado com o prefixo das primeiras j letras de R de forma a obter o prefixo das primeiras $i+j$ letras de T, então a tabela guarda o valor 1. Caso contrário, guarda o valor 0.

Complete a fórmula da recursão para a resolução deste problema. Considere que S_i denota o caracter na posição i da string S.

$$\texttt{M[i,j]} = \begin{cases} \boxed{} & \text{, se } i = 0 \text{ e } j = 0 \\ \boxed{} & \text{, se } (0 = i \text{ ou } S_i \neq T_{i+j}) \text{ e } (0 = j \text{ ou } R_j \neq T_{i+j}) \text{ e } (0 < i \text{ ou } 0 < j) \\ \boxed{} & \text{, se } (0 < i \text{ e } S_i = T_{i+j}) \text{ e } (0 = j \text{ ou } R_j \neq T_{i+j}) \\ \boxed{} & \text{, se } (0 = i \text{ ou } S_i \neq T_{i+j}) \text{ e } (0 < j \text{ e } R_j = T_{i+j}) \\ \boxed{} & \text{, se } (0 < i \text{ e } S_i = T_{i+j}) \text{ e } (0 < j \text{ e } R_j = T_{i+j}) \end{cases}$$

Solução:

$$\texttt{M[i,j]} = \begin{cases} \boxed{1} & \text{, se } i = 0 \text{ e } j = 0 \\ \boxed{0} & \text{, se } (0 = i \text{ ou } S_i \neq T_{i+j}) \text{ e } (0 = j \text{ ou } R_j \neq T_{i+j}) \text{ e } (0 < i \text{ ou } 0 < j) \\ \boxed{M[i-1,j]} & \text{, se } (0 < i \text{ e } S_i = T_{i+j}) \text{ e } (0 = j \text{ ou } R_j \neq T_{i+j}) \\ \boxed{M[i,j-1]} & \text{, se } (0 = i \text{ ou } S_i \neq T_{i+j}) \text{ e } (0 < j \text{ e } R_j = T_{i+j}) \\ \boxed{\max\{M[i-1,j], M[i,j-1]\}} & \text{, se } (0 < i \text{ e } S_i = T_{i+j}) \text{ e } (0 < j \text{ e } R_j = T_{i+j}) \end{cases}$$

III. (2,0 + 2,0 = 4,0 val.)

III.a) Considere o autómato finito determinista para o padrão $P = bababba$, indique os seguintes valores de transição: $\delta(4,b)$, $\delta(5,a)$, $\delta(6,b)$, $\delta(7,a)$, $\delta(7,b)$.

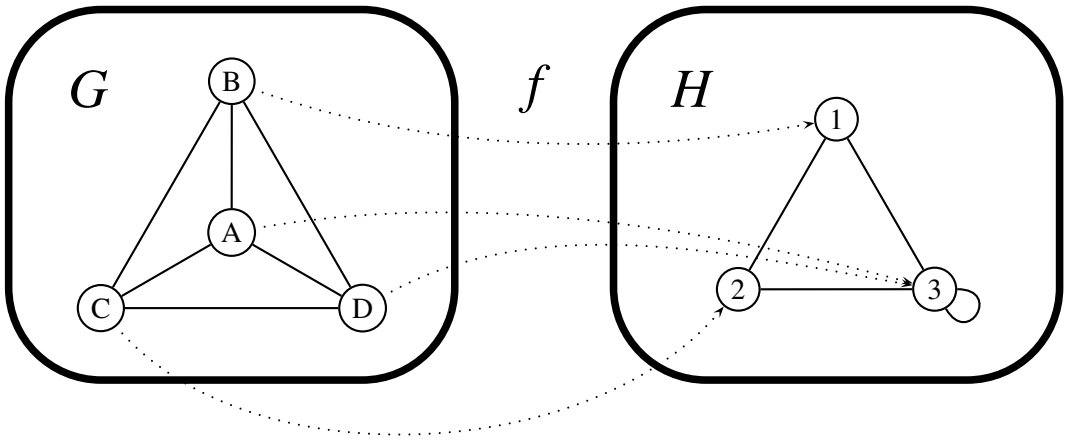
III.b) Calcule a função de prefixo do algoritmo de Knuth-Morris-Pratt para o padrão $P = aabaaabaab$.

IV. (2,0 + 2,0 = 4,0 val.)

IV.a) O Professor Carlos é um investigador perspicaz, que estuda o problema HORN-SAT e acabou publicar um algoritmo polinomial para resolver este problema. Assumindo que o algoritmo do Professor Carlos é correto classifique as seguintes afirmações como verdadeira (**V**), falsa (**F**) ou se não se sabe (**D**).

a. 2CNF-SAT $\in$ NP
b. 2CNF-SAT $\in$ NP-HARD
c. CLIQUE $\in$ NP-HARD
d. CLIQUE $\leq_P$ 3CNF-SAT
e. 3CNF-SAT $\leq_P$ CLIQUE

IV.b) Nesta questão vamos considerar o problema do homomorfismo de grafos não dirigidos (**GHmorphism**). Dados os grafos $G = (V_G, E_G)$ e $H = (V_H, E_H)$. Um homomorfismo é uma função $f : V_G \mapsto V_H$ que mapeia os vértices por forma a que os arcos também sejam mapeados em arcos. Se $\{u,v\}$ é um arco de G então $\{f(u), f(v)\}$ é um arco de H, i.e., se $\{u,v\} \in E_G$ então $\{f(u), f(v)\} \in E_H$. A figura seguinte ilustra um homomorfismo.



Os grafos são definidos com $V_G = \{A,B,C,D\}$, $E_V = \{\{A,B\},\{A,C\},\{A,D\},\{B,C\},\{B,D\},\{C,D\}\}$, $V_H = \{1,2,3\}$ e $E_H = \{\{1,2\},\{1,3\},\{2,3\},\{3,3\}\}$. Existe um homomorfismo entre estes dois grafos que efetua os seguintes mapeamentos: $f(B) = 1$, $f(C) = 2$, $f(A) = f(D) = 3$.

Dados dois grafos não dirigidos G e H o problema do homomorfismo (**GHmorphism**) consiste em determinar se existe algum homomorfismo de G para H. Na instância exemplificada a resposta era afirmativa, contudo caso o arco $\{3,3\}$ fosse retirado do grafo H então a resposta seria negativa.

Dado um grafo não dirigido $G = (V_G, E_G)$ o problema **3-COLOR** consiste em determinar se existe um forma de colorir os vértices de G, por forma a que vértices adjacentes não partilhem a mesma cor, usando apenas 3 cores diferentes. Formalmente o problema consiste em determinar se existe uma função $c : V_c \mapsto \{1,2,3\}$ tal que se $\{u,v\} \in E_G$ então $c(u) \neq c(v)$.

Sabendo que o problema **3-COLOR** é NP-Completo, prove que o problema **GHmorphism** é NP-Completo. Prove primeiro que **GHmorphism** $\in$ NP.

Solução:

Em primeiro lugar é necessário provar que **GHmorphism** $\in$ NP. Assumimos que são dados G, H e f, tais que G é representado em lista de adjacências, H é representado em matriz de adjacências e f é representado numa array indexada por V_G. Basta percorrer todos os elementos em $\{u,v\} \in E_G$ e verificar se o arco $\{f(u), f(v)\}$ existe na matriz de adjacência de H. Caso algum dos testes falhe o algoritmo returna falso, caso contrário retorna verdadeiro. Este algoritmo demora tempo $O(|V_H|^2 + |E_G| + |V_G|)$, pelo que é polinomial no tamanho do input.

Em segundo lugar vamos provar que **GHmorphism** $\in$ NP-HARD, fazendo uma redução a partir do **3-COLOR**. Utilizaremos para grafo G do **GHmorphism** o mesmo grafo G que é dado no input do **3-COLOR**. Precisamos também de definir o grafo H a utilizar no **GHmorphism**, o grafo H será o grafo H dado no enunciado, retirando o arco $\{3,3\}$.

Para verificarmos que esta redução está correcta temos que provar que a instância do **3-COLOR** tem solução se e só se a instância do **GHmorphism** gerada tem solução. A título de exemplo note que não é possivel colorir o grafo G no enunciado e também não existe homomorfismo para o grafo H modificado.

Se a instância do **3-COLOR** tem solução temos uma função de coloração $c : V_G \mapsto \{1,2,3\}$. Utilizamos esta função como sendo o homomorfismo f. Precisamos apenas de verificar que para qualquer $\{u,v\} \in E_G$ temos $\{c(u), c(v)\} \in E_H$, o que se verifica porque $c(u) \neq c(v)$.

No outro sentido queremos verificar que se a instância gerada do **GHmorphism** tem solução então a instância do **3-COLOR** original também tem solução. Escolhemos para função de coloração o homomorfismo h encontrado. Precisamos apenas de verificar que se $\{u,v\} \in E_G$

então $f(u) \neq f(v)$. Como f é um homomorfismo temos que $\{f(u), f(v)\} \in E_H$, mas como H não contem loops, dado que o arco $\{3,3\}$ foi retirado, temos que $f(u) \neq f(v)$.