



2º Exame, 3 de fevereiro de 2017, 11h30m Duração: 3 horas
Prova escrita, individual e sem consulta

NOME: _____ NÚMERO: _____

PARTE I - Questões de Escolha Múltipla (A/B/C/D)

Preencha as respostas na tabela (usando apenas letras maiúsculas). Se nenhuma opção servir, escreva **NENHUMA**. Se pretender alterar a sua resposta, risque e escreva ao lado a sua nova opção. Todas as questões de escolha múltipla seguintes valem 0.75 valores. Estas questões de escolha múltipla não respondidas são cotadas com 0 valores, mas por cada resposta errada são descontados 0.75/4 valores.

Questão	1	2	3	4	5	6
Resposta						

1. Considere a tabela indicada na 1a linha abaixo. As linhas seguintes correspondem, cada uma delas, ao resultado da aplicação de dois passos de um dado algoritmo de ordenação.

T	22	18	19	16	11	14	15	14	12	13
S1	11	12	19	16	22	14	15	14	18	13
S2	18	19	22	16	11	14	15	14	12	13
S3	11	12	22	18	19	16	13	14	15	14

Indique das seguintes correspondências entre tabelas (S1, S2 e S3) e algoritmos, qual é a correcta.

- | | |
|---|---|
| A. S1 - <i>Selection</i> , S2 - <i>Insertion</i> , S3 - <i>Bubble</i> | B. S1 - <i>Selection</i> , S2 - <i>Bubble</i> , S3 - <i>Insertion</i> |
| C. S1 - <i>Insertion</i> , S2 - <i>Selection</i> , S3 - <i>Bubble</i> | D. S1 - <i>Bubble</i> , S2 - <i>Insertion</i> , S3 - <i>Selection</i> |

2. Suponha que o algoritmo de união rápida ponderada com compressão de caminho é aplicado à seguinte sequência de pares de elementos:

6 - 7, 0 - 1, 1 - 7, 4 - 5, 4 - 0, 6 - 5

Qual das respostas indica correctamente: o número de uniões (U); e se no processo de compressão, **depois de** cada união, existe alguma alteração na tabela no ($C > 0$ se sim, $C = 0$ se não):

- | | | | |
|-------------------|-------------------|-------------------|-------------------|
| A. $U = 5, C > 0$ | B. $U = 6, C > 0$ | C. $U = 5, C = 0$ | D. $U = 6, C = 0$ |
|-------------------|-------------------|-------------------|-------------------|

3. Considere o seguinte acervo ("heap"), no qual a prioridade mais elevada corresponde ao menor valor.

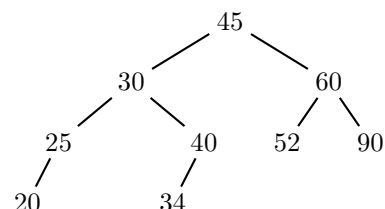
1	7	2	8	8	5	3	9
---	---	---	---	---	---	---	---

Qual o acervo obtido após serem realizadas as seguintes operações: i) inserir um novo elemento com prioridade 4 no acervo; ii) alterar a prioridade do primeiro elemento da tabela para 6; iii) retirar o elemento com maior prioridade.

Nota: após cada operação deve ser reposta a condição de acervo executando operações adequadas.

- | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A. <table><tr><td>2</td><td>7</td><td>3</td><td>8</td><td>8</td><td>5</td><td>4</td><td>9</td></tr></table> | 2 | 7 | 3 | 8 | 8 | 5 | 4 | 9 | B. <table><tr><td>3</td><td>4</td><td>5</td><td>8</td><td>7</td><td>8</td><td>6</td><td>9</td></tr></table> | 3 | 4 | 5 | 8 | 7 | 8 | 6 | 9 |
| 2 | 7 | 3 | 8 | 8 | 5 | 4 | 9 | | | | | | | | | | |
| 3 | 4 | 5 | 8 | 7 | 8 | 6 | 9 | | | | | | | | | | |
| C. <table><tr><td>3</td><td>4</td><td>5</td><td>7</td><td>8</td><td>8</td><td>6</td><td>9</td></tr></table> | 3 | 4 | 5 | 7 | 8 | 8 | 6 | 9 | D. <table><tr><td>4</td><td>6</td><td>3</td><td>7</td><td>8</td><td>5</td><td>8</td><td>9</td></tr></table> | 4 | 6 | 3 | 7 | 8 | 5 | 8 | 9 |
| 3 | 4 | 5 | 7 | 8 | 8 | 6 | 9 | | | | | | | | | | |
| 4 | 6 | 3 | 7 | 8 | 5 | 8 | 9 | | | | | | | | | | |

4. Considere a árvore binária ordenada, balanceada AVL abaixo. Supondo que nessa árvore se procede à inserção do número 32, indique qual das afirmações é verdadeira.



- A. A árvore permanece balanceada.
 B. É necessário executar uma rotação simples à esquerda no vértice 30.
 C. É necessário executar uma rotação dupla à esquerda no vértice 30.
 D. É necessário executar uma rotação simples à direita no vértice 40.

5. Considere um conjunto de elementos que são armazenado numa tabela de dispersão com índices livres, em que a resolução de colisões é feita por procura linear. A tabela seguinte indica para um subconjunto desses elementos, a chave de cada elemento e o resultado da chamada à função de dispersão:

chave	A	B	C	D	E	F	G	H	...
hash	0	2	5	1	4	1	3	0	...

Considere as seguintes tabelas e indique qual delas poderia corresponder ao topo da tabela de dispersão se os elementos forem introduzidos por uma determinada ordem (o símbolo • significa que a posição correspondente está vazia):

A.	<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>...</td></tr><tr><td>A</td><td>F</td><td>B</td><td>G</td><td>E</td><td>C</td><td>•</td><td>D</td><td>...</td></tr></table>	0	1	2	3	4	5	6	7	...	A	F	B	G	E	C	•	D	...	B.	<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>...</td></tr><tr><td>A</td><td>D</td><td>B</td><td>C</td><td>F</td><td>E</td><td>H</td><td>•</td><td>...</td></tr></table>	0	1	2	3	4	5	6	7	...	A	D	B	C	F	E	H	•	...
0	1	2	3	4	5	6	7	...																															
A	F	B	G	E	C	•	D	...																															
0	1	2	3	4	5	6	7	...																															
A	D	B	C	F	E	H	•	...																															
C.	<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>...</td></tr><tr><td>A</td><td>E</td><td>C</td><td>H</td><td>B</td><td>D</td><td>F</td><td>G</td><td>...</td></tr></table>	0	1	2	3	4	5	6	7	...	A	E	C	H	B	D	F	G	...	D.	<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>...</td></tr><tr><td>H</td><td>D</td><td>F</td><td>B</td><td>G</td><td>E</td><td>C</td><td>A</td><td>...</td></tr></table>	0	1	2	3	4	5	6	7	...	H	D	F	B	G	E	C	A	...
0	1	2	3	4	5	6	7	...																															
A	E	C	H	B	D	F	G	...																															
0	1	2	3	4	5	6	7	...																															
H	D	F	B	G	E	C	A	...																															

6. Considere a seguinte recorrência

$$C_N = C_{N-1} + N \log(N)$$

Qual dos seguintes conjuntos representa a complexidade temporal associada à recorrência dada?

- A. $\mathcal{O}(N \log(N))$ B. $\mathcal{O}(N \log^2(N))$ C. $\mathcal{O}(N^2)$ D. $\mathcal{O}(N^2 \log(N))$

PARTE II - Questões de Escolha Binária (V/F)

Preencha as respostas na tabela (usando apenas letras maiúsculas – V(erdadeira) ou F(alsa)). Todas as questões de escolha múltipla seguintes valem 0.50 valores. Estas questões de escolha múltipla não respondidas ou erradas são cotadas com 0 valores.

Questão	7	8	9	10	11	12	13	14
Resposta								

7. O algoritmo de ordenação Heapsort tem no pior caso uma complexidade $\mathcal{O}(N^2)$.
 8. Numa lista duplamente ligada com 11 elementos, para retirarmos o elemento do meio da lista necessitamos de **alterar** 2 apontadores na estrutura da lista.
 9. Dado um qualquer item num acervo, que não esteja na raiz do mesmo, aproxima-se sempre da raiz quando a sua prioridade aumenta.
 10. A maior aresta de um grafo nunca faz parte de uma qualquer SPT construída sobre esse grafo.

11. Se $\lim_{N \rightarrow \infty} \frac{Nf(N)}{g(N)} < \infty$, então pode-se afirmar que $f(N) \in \mathcal{O}(g(N))$.
12. Relativamente ao problema da conectividade, a tabela `id = [1 1 1 ... 1]` nunca pode ser obtida quando se aplica o algoritmo da união rápida.
13. A procura de um elemento numa tabela ordenada pode ser feita em $\mathcal{O}(\log N)$ operações.
14. Uma árvore binária com k níveis (de zero a $k - 1$) tem, no máximo, $2^k - 1$ elementos.

PARTE III - Questões de Desenvolvimento

Responda a cada uma das questões de desenvolvimento em **folhas de exame separadas** e devidamente identificadas com nome e número.

[5.0]

15. Dada uma string de caracteres, é possível produzir outras strings que correspondam a permutações dos caracteres da string inicial. Por exemplo, a partir da string “ahd” podem produzir-se cinco strings diferentes: “adh”, “had”, “hda”, “dah” e “dha”. O que se pretende neste exercício é desenvolver um algoritmo, e uma forma de representar informação relevante, que permita identificar eficientemente se duas strings são ou não alguma permutação uma da outra.

Por exemplo, se as strings dadas forem “abcd” e “badc”, o resultado deverá ser afirmativo, mas se forem “abcd” e “abbc”, ou “abcd” e “abc”, o resultado deverá ser negativo. As strings a processar podem ter qualquer tamanho, mas apenas quando ambas as strings têm igual tamanho é que o resultado poderá ser afirmativo, sendo sempre negativo se os tamanhos forem diferentes.

A preocupação com eficiência neste problema prende-se com a possibilidade de se ter que processar strings de comprimento elevado. É admissível ter-se que processar strings com dezenas ou centenas de milhar de caracteres. Admite-se que existem apenas 8 bits para representar os caracteres, pelo que o número máximo de caracteres utilizáveis é 256. Uma qualquer string poderá ter qualquer desses caracteres bem como poderá ter múltiplas ocorrências de cada um deles.

- (2.0) (a) Proponha um algoritmo para resolver o problema, descrevendo-o de forma clara e sem ambiguidades de interpretação, e ilustre como o seu algoritmo garantidamente resolve o problema correctamente. Poderá ser útil traçar um fluxograma, assim como descrever com detalhe alguma estrutura de dados adicional que entenda relevante para a resolução do problema. Note que os dados de entrada do problema serão apenas dois ponteiros para `char`.

Nota: A cotação atribuída será dependente da eficiência do algoritmo proposto, pelo que deverá tentar produzir o algoritmo mais eficiente possível, não bastando apenas propor um algoritmo que resolva o problema (há vários possíveis).

- (1.0) (b) Demonstre de forma justificada qual a complexidade do algoritmo por si proposto na alínea anterior. Na sua análise não se esqueça que as strings podem ter tamanhos elevados, mas também podem ter tamanhos reduzidos.

- (2.0) (c) Escreva em linguagem C a função `int IsPermutation(char * s1, char * s2)`; que implemente o algoritmo por si proposto. A função deverá devolver 1 quando `s2` for uma permutação de `s1` e deverá devolver 0 no caso contrário. Confirme e mostre que a função por si escrita possui de facto a complexidade que indicou na alínea anterior.

Nota: Para simplificar a implementação, a função deverá também devolver 1 se `s1` e `s2` forem strings iguais.

IDENTIFIQUE A FOLHA DE EXAME E MUDE DE FOLHA !!!

- [4.5] 16. Considere um grafo ponderado, com 8 vértices identificados de A a H, representado pela matriz de adjacências indicada abaixo.

	A	B	C	D	E	F	G	H
A		2	2					
B	2		3		1	3		
C	2	3		2				
D			2		4			1
E		1		4			1	5
F		3					1	
G					1	1		2
H				1	5		2	

- [1.75] a) Determine a árvore de caminhos mais curtos (SPT) tomando o vértice **B** como fonte. Apresente os seus cálculos de forma clara, detalhada e completa para cada iteração do algoritmo. Por exemplo, mas sem se restringir a estes aspectos, identifique a franja da procura e pesos, assim como deverá indicar por que ordem entra cada vértice na árvore e o estado da franja em cada momento.
- [0.25] b) Indique quais os vértices que fazem parte do caminho mais curto entre o vértice **B** e o vértice **H**, bem como qual o seu valor de custo.
- [1.0] c) Seja P o caminho mais curto entre os nós s e t num dado grafo G . Se o peso de todas as arestas do grafo for aumentado por uma unidade, P continua a ser o caminho mais curto de s para t ? Responda justificadamente e se considerar que a resposta é não, dê um contra exemplo.
- [1.5] d) Considere os vectores st e wt indicados abaixo que se obtém após algumas iterações de um algoritmo para determinação da árvore de caminhos mais curtos (SPT) (Note que não se trata do resultado final mas sim do estado das tabelas numa fase intermédia do algoritmo). Diga, justificando, quais os vértices que nesta fase já estão **necessariamente** na SPT final.

st	7	7	9	2	1	4	7	2	7	1	9	4	9
wt	9	4	11	13	25	17	0	19	6	31	24	29	27

IDENTIFIQUE A FOLHA DE EXAME E MUDE DE FOLHA !!!

- [2.0] 17. A determinação dos K primeiros elementos de um conjunto, ou equivalentemente do k -ésimo numa dada ordem, é um problema de interesse em muitos casos. Um desses casos é por exemplo a determinação da mediana de um conjunto de elementos, mas existem muitas outras situações práticas em que é relevante ter uma solução eficiente para este problema. Dada uma tabela $T[]$ de N números e uma tabela $R[]$ com K posições ($N > K$), pretende-se encontrar os K primeiros elementos de $T[]$ e retorná-los nas K posições de $R[]$. Uma solução para este problema é fazer K passos de “Selection Sort” numa tabela de N elementos e copiar então os primeiros K elementos para $R[]$. No entanto a complexidade dessa operação, se $K \approx \mathcal{O}(N)$ é quadrática ($\mathcal{O}(N^2)$) o que para N grande é muito penalizador.
- (0.5) (a) Indique uma forma mais eficiente de encontrar os K menores elementos de $T[]$ e retorná-los em $R[]$. Justifique a sua resposta.
Nota: Não necessita escrever código mas explique com clareza o algoritmo que sugere.
- (0.5) (b) Qual é a complexidade do algoritmo que propôs em função de K e N e em particular se $K \approx \mathcal{O}(N)$.
- (0.5) (c) É possível resolver o problema da selecção com complexidade linear em N ($K \approx \mathcal{O}(N)$). Descreva um algoritmo para esse efeito justificando detalhadamente todos os passos.
Nota: Não necessita escrever código mas explique com clareza o algoritmo que sugere.
- (0.5) (d) Justifique que o algoritmo proposto tem de facto complexidade linear em função de N .