



1º Exame, 14 de Julho de 2021, 15h00m 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	7	8
Resposta	B							

1. No contexto do problema da conectividade, a tabela `id[.]` após algumas iterações encontra-se como indicado abaixo. Nessa tabela existem três conjuntos de raízes, respectivamente 2, 7 e 10.

10	9	2	7	6	10	2	7	0	7	10	0
----	---	---	---	---	----	---	---	---	---	----	---

Qual é a afirmação verdadeira?

- A. Nenhum dos três conjuntos é consistente com a utilização do algoritmo da união rápida ponderada.
- B. O conjunto de raiz 2 é inconsistente com a utilização do algoritmo da união rápida ponderada, mas o conjunto de raiz 10 é.
- C. O conjunto de raiz 7 é consistente com a utilização do algoritmo da união rápida ponderada, mas o conjunto de raiz 10 é inconsistente.
- D. Todos os conjuntos são consistentes com a utilização do algoritmo da união rápida ponderada.

2. Uma tabela de dispersão de índices livres de tamanho  $M = 100$  inicialmente vazia recebeu um conjunto de elementos. O segmento da tabela com início na posição de índice 25 e final na posição de índice 31 está assim preenchido:

...	2324	1326	3827	228	429	729	4225	...
-----	------	------	------	-----	-----	-----	------	-----

Qual das sequências abaixo produz aquele preenchimento se as colisões se resolverem por **procura linear**?

- A. (429; 1326; 4023; 228; 723; 2324; 729; 3827; 4225)
- B. (1326; 723; 429; 3827; 4023; 228; 729; 4225; 2324)
- C. (228; 4023; 1326; 723; 2324; 429; 3827; 4225; 729)
- D. (723; 4023; 2324; 3827; 1326; 429; 729; 228; 4225)

3. Para um dado grafo não direccionado representado por matriz de adjacências aplicou-se **DFS** para a determinação e um caminho simples entre dois vértices. O caminho produzido foi

$$6 - 3 - 0 - 7 - 4 - 2 - 9$$

Sabendo que o caminho foi obtido sem que o procedimento recuasse, qual dos seguintes caminhos de certeza não poderia ser obtido noutra invocação da **DFS**?

- A. 9 - 0 - 3 - 2      B. 1 - 3 - 0 - 7      C. 2 - 4 - 6 - 3      D. 4 - 0 - 3 - 5

4. Suponha um conjunto de  $N$  números inteiros ( $N$  muito grande), organizados sobre os quais é implementado o seguinte algoritmo: dado um número  $k$ , esse número é pesquisado e o algoritmo retorna TRUE ou FALSE respectivamente, consoante encontrou ou não o número  $k$ . Em média **quantas comparações** são efectuadas durante a execução do algoritmo **sendo este o mais eficiente possível**?

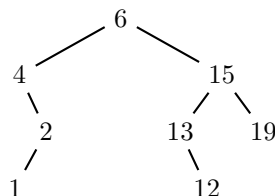
- A.  $\mathcal{O}(N)$  se os dados forem representados em tabela ordenada ou não.  
 B.  $\mathcal{O}(\log N)$  se os dados forem representados em lista ordenada.  
 C.  $\mathcal{O}(\log N)$  se os dados forem representados em tabela ordenada.  
 D.  $\mathcal{O}(N)$  se os dados forem representados em árvore ordenada **AVL**.

5. Considere a seguinte tabela (1ª linha) sobre a qual são listados alguns passos executados pelo algoritmo Quick Sort. Cada linha a seguir à primeira corresponde ao resultado após a conclusão de um passo do algoritmo, qual será o resultado do próximo passo do algoritmo?

9	1	19	4	20	6	34	30	37	73
9	1	19	4	20	6	34	30	37	73
9	1	19	4	20	6	34	30	37	73
9	1	19	4	20	6	30	34	37	73
9	1	19	4	20	6	30	34	37	73
4	1	6	19	20	9	30	34	37	73

- A. | 1 | 4 | 6 | 9 | 19 | 20 | 30 | 34 | 37 | 73 |      B. | 1 | 4 | 6 | 9 | 20 | 19 | 30 | 34 | 37 | 73 |  
 C. | 4 | 1 | 6 | 9 | 19 | 20 | 30 | 34 | 37 | 73 |      D. | 4 | 1 | 6 | 9 | 20 | 19 | 30 | 34 | 37 | 73 |

6. Dada a árvore binária que abaixo se apresenta, qual das seguintes afirmações é falsa?



- A. O varrimento pré-fixado produz: 6-4-2-1-15-13-12-19.  
 B. O varrimento in-fixado produz: 4-1-2-6-13-12-15-19.  
 C. O varrimento pós-fixado produz: 1-2-4-13-12-19-15-6.  
 D. A árvore não é balanceada AVL

7. Considerando o acervo abaixo, em que maior valor corresponde a maior prioridade.

73	49	34	29	17	11	16	1	23	5
----	----	----	----	----	----	----	---	----	---

Qual das seguintes tabelas representa o acervo que resulta da aplicação das seguintes operações: inserir o elemento 50; remover o elemento de maior prioridade; modificar a prioridade do elemento 34 para 2.

A.	50	49	16	29	17	11	2	1	23	5
B.	50	49	16	29	17	2	11	1	23	5
C.	50	49	16	29	23	11	2	1	17	5
D.	50	49	16	29	23	2	11	1	17	5

8. Qual das seguintes afirmações é **falsa**?

A.	$f(N) = N \lg(N^2) \in \mathcal{O}(N \lg(N))$	B.	$f(N) = \lg(N \lg(N)) \in \mathcal{O}(\lg(N))$
C.	$f(N) = N \lg^2(N) \in \mathcal{O}(N^2)$	D.	$f(N) = N^2 \lg(N) \in \mathcal{O}(N \lg^2(N))$

## 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	9	10	11	12	13	14	15	16
Resposta								

9. Um programa que usa um tipo abstracto é um cliente. O cliente tem acesso à implementação, logo se a implementação for alterada será necessário actualizar a definição da respectiva interface para o cliente.
10. Num vector ordenado ao contrário os algoritmos selection sort e quick sort produzem exactamente os mesmos resultados após cada passo do algoritmo.
11. Uma árvore binária com k níveis (de zero a k - 1) tem, no máximo,  $\frac{k(k+1)}{2}$  elementos.
12. Se forem inseridos os elementos 26, 4 e 13 sucessivamente, por esta ordem, no acervo representado em baixo são efectuadas 0 trocas.

50	37	34	29	17	11	16	1
----	----	----	----	----	----	----	---

13. Seja uma tabela de dispersão de índices livres com resolução de colisões através de procura linear e seja uma outra tabela de dispersão com resolução de colisões por dispersão em lista. Se ambas as tabelas possuírem a mesma dimensão e se forem preenchidas com os mesmos dados, o tamanho médio dos clusters da primeira é estatisticamente igual ao tamanho médio das listas da segunda.
14. Um grafo que não possua ciclos ou é uma árvore ou uma floresta.
15. Embora possua complexidade assintótica equivalente à da remoção do elemento mais prioritário em listas ordenadas, a remoção do elemento mais prioritário em tabelas ordenadas é, em geral, sempre mais lenta.
16. A técnica de projecto designada como Programação Dinâmica Ascendente é adequada para problemas que não geram subproblemas independentes, enquanto a Programação Dinâmica Descendente não é.

## 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.

- [4.0] 17. Considere a matriz de adjacências abaixo que representa um grafo ponderado direccionado de 11 vértices, numerados de 0 a 10.

	0	1	2	3	4	5	6	7	8	9	10
0	0		3			5			8		
1	4	0			6		3			2	
2	5		0	6				7			2
3				0	2		3		5		7
4	6			7	0			1			
5			8			0	4	3			2
6				3	8	7	0				
7		6	2				1	0			9
8			9	4			8		0	4	
9			4		2		7			0	10
10	4		6							6	0

- [2.0] a) Determine a árvore de caminhos mais curtos com fonte no vértice 0. Justifique os seus cálculos e trace a árvore final, identificando claramente as distâncias de cada vértice ao vértice fonte.
- [1.0] b) O algoritmo de Dijkstra diz-se ser linear para grafos representados por matrizes de adjacências, mas o mesmo não se pode dizer quando se aplica o mesmo algoritmo a grafos representados por listas de adjacências. Porquê? Aproveite para explicitar qual a sua interpretação do que é um algoritmo ser linear em grafos.
- [1.0] c) Para um grafo de 12 vértices ponderado e direccionado aplicaram-se algumas iterações do algoritmo de Dijkstra, após o que se registaram os valores presentes no vector `st[.]` e no vector `wt[.]`. Os valores observados foram os seguintes:

$st = [4; 6; 9; 3; 3; 6; 3; 4; 9; 6; 5; 5]$   
 $wt = [12; 11; 13; 0; 8; 7; 5; 16; 12; 10; 9; 16]$

Por observação daqueles dois vectores indique quais são os vértices que seguramente já pertencem à árvore final e quais os vértices que seguramente ainda não fazem parte dela. Existem vértices relativamente aos quais não é possível dizer se já pertencem ou ainda não pertencem à árvore final? Quais? Justifique as suas afirmações fundamentadamente.

- [4.0] 18. Uma outra estrutura para representação de grafos que é mais eficiente para representação de grafos esparsos é o *compressed sparse row*. Esta estrutura permite representar grafos esparsos utilizando apenas tabelas e gastando apenas espaço  $\mathcal{O}(V + E)$  para um grafo com  $V$  vértices e  $E$  arestas. Considere o exemplo abaixo em que se representa o mesmo grafo utilizando uma matriz de adjacências e um *compressed sparse row*.

	0	1	2	3	4	5
0	0		1	3	5	
1		0	4	7		
2	3	1	0			2
3	2			0		1
4		4			0	
5			2		1	0

Índices das Linhas

0	1	2	3	4	5	fim
0	3	5	8	10	11	13

Índices das Colunas

2	3	4	2	3	0	1	5	0	5	1	2	4
---	---	---	---	---	---	---	---	---	---	---	---	---

Vector de Pesos

1	3	5	4	7	3	1	2	2	1	4	2	1
---	---	---	---	---	---	---	---	---	---	---	---	---

Nesta representação são utilizados três vectores. O primeiro destes é utilizado para guardar os índices das linhas, ou seja, a posição do vector de índices das colunas em que começam as arestas

que partem do respectivo vértice. O segundo vector guarda os índices das colunas para cada aresta, note-se que estes estão por ordem e que são guardadas primeiro as arestas que partem do vértice 0, depois do vértice 1, do vértice 2 e assim sucessivamente. Isto para que utilizando o primeiro vector seja possível aceder directamente a cada aresta com complexidade  $\mathcal{O}(\deg(v))$ , ou seja com tempo proporcional ao grau do vértice de partida. O terceiro vector guarda os pesos para cada aresta, sendo acedido do mesmo modo que o segundo.

Ou seja com esta estrutura por exemplo se pretendermos aceder à aresta  $2 \mapsto 5$ , temos que aceder ao primeiro vector na posição  $2 + 2 + 1$ , sendo possível verificar que as arestas no segundo vector estão entre a posição 5 e a posição  $8 - 1$ . Depois de se procurar nestas posições o valor 5, basta aceder à respectiva posição no vector de pesos e temos o peso da aresta correspondente. Caso não se encontrasse o valor 5 entre as posições referidas é porque a aresta  $2 \mapsto 5$  não existia.

Nas alíneas abaixo considere a seguinte definição para o tipo referente ao *compressed sparse row*.

```
typedef struct _CSR
{
    int V;
    int *rowindex;
    int *colindex;
    int *weights;
}CSR;
```

- [1.0] a) Produza código em linguagem C que implemente a função com a seguinte assinatura:

**int getDegree(CSR\* graph, int v)**

em que a mesma retorna o grau de saída do vértice  $v$  no grafo  $graph$ .

- [1.5] b) Produza código em linguagem C que implemente a função com a seguinte assinatura:

**CSR\* getCSR(int\*\* adjm, int v)**

em que a mesma retorna um *compressed sparse row* com o grafo com  $v$  vértices representado em  $adjm$  libertando de seguida a memória correspondente à matriz de adjacências.

- [1.5] c) **Sem escrever código**, descreva, **em menos de 15 linhas**, como poderia implementar uma função com a seguinte assinatura:

**int isDirectedDegrees(CSR\* graph)**

em que a mesma retorna 1 caso verifique que o grafo é direccionado e 0 caso não consiga garantir que o mesmo é direccionado. Esta função deve recorrer apenas à determinação do grau de entrada e de saída de cada vértice para produzir o resultado pretendido.

- [2.0] 19. Neste problema pretende-se proceder à construção de uma árvore binária ordenada e balanceada AVL

- [1.5] a) Assuma a árvore inicialmente vazia onde se inserem os valores seguintes, pela ordem apresentada: 58, 67, 73, 62, 60, 14, 64, 7.

Após cada inserção a árvore deverá ser balanceada AVL. Assim, cada nova inserção deverá indicar se a árvore permanece balanceada ou não. Nos casos em que não fique balanceada, deverá indicar qual a rotação a fazer - se simples, se dupla, se à direita, se à esquerda e em que vértice se realiza -, e executar a rotação por si indicada.

Na sua resolução deverá apresentar a árvore imediatamente após cada inserção e, quando tal for o caso, apresentar a árvore que resulta da rotação executada. Só depois deverá prosseguir com as restantes inserções.

- [0.5] b) Na alínea anterior foram efectuadas operações de inserção e como resultado de algumas dentre elas, foi necessário alterar ligações na árvore para a deixar balanceada. No pior caso quantas operações de balanceamento são efectuadas após uma inserção?