DEEC
DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

AED - 2020-21 - 2º Semestre
Algoritmos e Estruturas de Dados

1^o Exame, 30 de Junho de 2021, 18h00m 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. Suponha que, aplicando o algoritmo da união rápida ponderada com compressão de caminho, logo após a última união e antes dos ciclos de compressão, a tabela `id[.]` resultante de se unir o par (4; 9) é a que se apresenta abaixo.

12	13	6	16	2	2	6	6	6	7	16	16	6	13	7	1	16	13	9

Qual das seguintes tabelas se obtém depois de concluídos os ciclos de compressão?

A.

12	13	6	16	6	2	6	6	6	6	16	16	6	13	7	1	16	13	6

B.

12	13	6	16	6	6	6	6	6	6	16	16	6	13	6	1	16	13	9

C.

12	13	6	16	6	2	6	6	6	6	16	16	6	13	7	1	16	13	9

D.

12	13	6	16	6	6	6	6	6	6	16	16	6	13	6	1	16	13	6

2. Dada uma estrutura de dados, para se remover **qualquer** elemento é necessário alterar 2 ponteiros. Que tipo de estrutura temos?

A. Lista simples. B. Lista duplamente ligada. C. Lista circular duplamente ligada. D. Árvore.

3. Para uma tabela de dispersão de índices livres inicialmente vazia e de tamanho $M = 100$ em que as colisões se resolvem por procura linear e considerando o segmento da tabela com início na posião de índice 15 e final na posição de índice 21, qual das tabelas é consistente com a seguinte sequência de chegada: (419; 1316; 4013; 218; 713; 2314; 719; 3817; 4215)

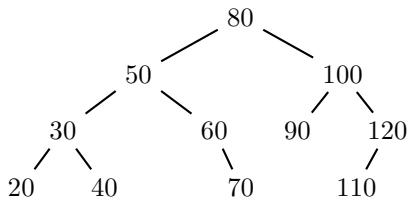
A. ... | 2314 | 1316 | 3817 | 218 | 419 | 719 | 4215 | ...

B. ... | 4215 | 1316 | 3817 | 218 | 419 | 719 | 2314 | ...

C. ... | 2314 | 1316 | 3817 | 218 | 419 | 4215 | 719 | ...

D. ... | 2314 | 1316 | 3817 | 4215 | 419 | 719 | 218 | ...

4. Dada a árvore binária ordenada e balanceada AVL qual é o conjunto que tem os elementos que podem ser inseridos **sucessivamente** e sem necessidade de operações de balanceamento



A. 128, 112, 63	B. 11, 25, 18	C. 67, 57, 127	D. 57, 67, 127.

5. Aplicou-se o algoritmo de Dijkstra num grafo não direccionado e ponderado, com todas as arestas não negativas. O vector `st[.]` que se obteve no final foi

$$\text{st} = [10;\ 7;\ 11;\ 7;\ 1;\ 1;\ 0;\ 7;\ 0;\ 11;\ 7;\ 3]$$

Indique qual dos vectores `wt[.]` abaixo é válido.

A. wt = [4; 4; 11; 6; 7; 9; 11; 0; 6; 11; 3; 5]	B. wt = [4; 4; 13; 6; 7; 9; 11; 0; 6; 7; 3; 8]
C. wt = [4; 4; 12; 6; 7; 3; 11; 0; 6; 11; 3; 8]	D. wt = [4; 4; 14; 6; 7; 9; 11; 0; 6; 11; 3; 8]

6. Considere a seguinte tabela (1ª linha) sobre a qual são listados alguns passos executados por um algoritmo de ordenação. Cada linha a seguir à primeira corresponde ao resultado após a conclusão de um passo do algoritmo, qual é o algoritmo utilizado?

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

A. Bubble Sort
B. Insertion Sort
C. Quick Sort
D. Selection Sort

7. Para uma função recursiva determinou-se que a recorrência que descreve a sua complexidade temporal em pior caso é

$$C_N = C_{N-1} + \lg N^2$$

Qual dos conjuntos abaixo constitui o menor majorante que corresponde àquela recorrência?

A. $C_N \in \mathcal{O}(\lg^2 N)$	B. $C_N \in \mathcal{O}(3N \lg N)$	C. $C_N \in \mathcal{O}(N^2)$	D. $C_N \in \mathcal{O}(N^2 \lg N)$

8. Tendo-se um acervo inicialmente vazio, é inserida a sequência de números: 17, 6, 7, 11, 13, 3, 2. Assumindo que a maior prioridade é dada aos menores números e que a inserção é feita no fim do acervo, quantos "fixUp"ou "fixDown"produzem alterações?

A. 3	B. 4	C. 5	D. 6

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								

9. Numa fila FIFO o primeiro elemento a sair é o primeiro elemento a entrar enquanto que numa pilha LIFO o primeiro elemento a sair é o último a entrar.

10. A complexidade do algoritmo quick sort é sempre $\mathcal{O}(N \log N)$.

11. Numa árvore binária ordenada o elemento que possua a menor chave pode ter filhos à sua direita.

12. A extração do elemento de maior prioridade de um acervo pode ser feita em tempo constante, mas a reposição da ordem própria do acervo implementada na sua forma mais eficiente é feita em tempo linear.

13. Para construir uma implementação completa de tabelas de dispersão basta escolher a dimensão da tabela e o seu formato (índices livres ou dispersão em lista) e o procedimento de resolução de colisões.

14. Qualquer grafo de V vértices que possua $V - 1$ arestas é uma árvore.

15. Qualquer grafo não ponderado em que o grau de saída de cada vértice (somatório das aresta que o têm como fonte) é igual ao seu grau de entrada (somatório das arestas que o têm como destino) é um grafo não direccionado.

16. Embora possua complexidade assimptótica equivalente à da inserção em listas ordenadas, a insersão em tabelas ordenadas é, em geral, sempre mais lenta.

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 o grafo ponderado não direccionado com 11 vértices, identificados de 0 a 10, representado pela matriz de adjacências indicada abaixo. A ausência de valor para um par linha e coluna significa a ausência de aresta entre esses dois vértices, i.e., custo ∞.

	0	1	2	3	4	5	6	7	8	9	10
0	0	3		4	6					10	19
1	3	0	3		7		9	10	6		
2		3	0	2	4	8			12		
3	4		2	0		9	11	8	7		
4	6	7	4		0	16		5		13	
5			8	9	16	0					5
6		9		11			0	18	7	2	6
7		10		8	5		18	0		4	3
8		6	12	7			7		0	1	8
9	10				13		2	4	1	0	
10	19					5	6	3	8		0

[2.0] a) Tomando o vértice 4 como ponto de partida determine a árvore mínima de suporte para este grafo. Justifique os cálculos de forma clara, assim como todas as decisões tomadas ao longo da determinação da árvore.

[1.0] b) Trace a árvore obtida na alínea anterior e apresente os vectores wt e st que a ela estão associados.

[1.0] c) Explique se pode ou não e porquê usar essa árvore para determinar o caminho mais curto entre os vértices 0 e 4. Podendo indique qual é o caminho mais curto.

[4.5] 18. A função `it_keeps_growing_and_growing`, que para efeitos de poupança de caracteres vamos chamá-la de `everest`, está definida para todos os inteiros positivos da seguinte forma:

$$\begin{aligned} \text{everest}(1) &= 1 \\ \text{everest}(2n) &= 2\text{everest}(n) \\ \text{everest}(2n+1) &= 2n+1+2\text{everest}(n)+\frac{1}{n}\text{everest}(n) \end{aligned}$$

É possível mostrar que everest(n) é inteiro para todo o valor de n.

[1.5] a) Escreva uma função em linguagem C que receba dois argumentos inteiros, `N` e `p`, e que devolva o valor de everest(N) mod p. O objectivo do segundo argumento é evitar que os valores devolvidos percam significado por excederem o maior inteiro representável. A função deverá ter a seguinte assinatura:

```
int everest(int N, int p);
```

Comente devidamente o seu código para que o algoritmo usado seja claro.

[0.5] b) Indique de forma justificada qual a complexidade temporal da função que implementou na alínea anterior. Caso se justifique, discuta qual a complexidade espacial da mesma.

[2.0] c) Suponha agora que se pretende calcular o valor $S(n)$ definido abaixo

$$S(n) = \sum_{i=1}^{n} [\text{everest}(n)]^2 \tag{1}$$

Escreva uma implementação da função

```
int sum_everest_squares(int N, int p);
```

que calcula aquele valor para N, apresentando o resultado em `modulo` p. Comente devidamente o seu código para que o algoritmo usado seja claro.

[0.5] d) Indique de forma devidamente fundamentada qual a complexidade temporal da função que implementou em c).

Nota1: Dados dois inteiros, a e p, se $a < p$, então a mod $p = a$ e se $a > p$, então a mod $p = (a - p)$ mod p. Adicionalmente, $(a + b)$ mod $p = [(a \bmod p) + (b \bmod p)]$ mod p. Também convém recordar que ab mod $p = [(a \bmod p)(b \bmod p)]$ mod p.

Nota2: Qualquer das funções pedidas admite implementações mais e menos eficientes e o que é bom para uma poderá não sê-lo para a outra. A arte está em acertar com a receita adequada em cada um dos casos.

[1.5] 19. Aplicando o Master Theorem às recorrências $C_N = C_{N/2} + N$ e $C_N = C_{N/3} + N$ obtém-se o mesmo resultado. No entanto, medindo experimentalmente, via estudo extensivo, o tempo gasto, observou-se que o segundo algoritmo termina sempre mais cedo que o primeiro.

Houve erro na recolha ou os dados experimentais estão certos? Se estão certos, como conciliar esta observação com o resultado do Master Theorem?