



1º Exame, 11 Junho 2013, 11:30h      Duração: 3 horas  
Prova escrita, individual e sem consulta

NOME: \_\_\_\_\_ NÚMERO: \_\_\_\_\_

### PARTE I - Questões de Escolha Múltipla

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 valem 0.75 valores. As 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								

1. Considere a seguinte tabela de indivíduos:

José	Maria	Alberto	António	Luís	Sofia	Laura	Telmo	Beatriz	Ricardo	Ana	Emília
------	-------	---------	---------	------	-------	-------	-------	---------	---------	-----	--------

Num ficheiro de dados são indicados pares destes indivíduos, em que cada par representa a existência de um qualquer parentesco entre os dois indivíduos. Após a aplicação do **algoritmo da união rápida** a todos os pares existentes no referido ficheiro, obteve-se a seguinte tabela

Ricardo	Maria	Alberto	Maria	Laura	Ricardo	Laura	Beatriz	Emília	Ricardo	António	Laura
---------	-------	---------	-------	-------	---------	-------	---------	--------	---------	---------	-------

Quantas famílias existem no ficheiro de dados?

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

2. Considere o seguinte trecho de código à esquerda. Assuma que a complexidade da função *swap* é  $O(1)$ . Indique qual dos conjuntos à direita reflete a complexidade computacional do código:

```
heapPerm(char * str, int n){
    int i;
    if (n == 0)
        printf("%s",str);
    else
        for (i = 0; i < n; i++) {
            heapPerm(str, n-1);
            if (n % 2)
                swap(str[0], str[n-1]);
            else
                swap(str[i], str[n-1]);
        }
}
```

- |                  |
|------------------|
| A. $O(N)$        |
| B. $O(N \log N)$ |
| C. $O(N^2)$      |
| D. $O(N!)$       |

3. Considere a seguinte tabela de números:

30	5	93	58	50	41	13	86	59	13	45	27
----	---	----	----	----	----	----	----	----	----	----	----

Indique das opções seguintes a tabela que se obtém por aplicação do **algoritmo "Quicksort"** após a 1ª partição, considerando como pivot o elemento mais à direita.

A.	5	13	13	27	30	41	45	50	58	59	86	93
B.	13	5	27	58	50	41	93	86	59	30	45	13
C.	5	13	13	27	41	50	93	86	59	30	45	58
D.	13	5	13	27	50	41	93	86	59	30	45	58

4. Considere uma fila com prioridade (em que mais prioritário possui valor numérico mais alto) baseada num **acervo** ternário, o qual é representado por uma árvore ternária perfeitamente balanceada em que os filhos do nó  $i$  estão nas posições  $3i-1$ ,  $3i$ ,  $3i+1$ . Suponha que tem a seguinte tabela que representa um acervo ternário:

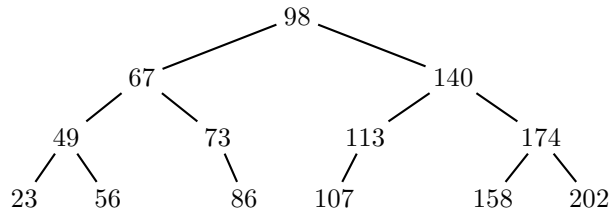
88	33	77	66	10	30	25	23	60	75	14	21	50	9	7
----	----	----	----	----	----	----	----	----	----	----	----	----	---	---

Indique das opções seguintes a tabela que se obtém ao remover o elemento mais prioritário da fila.

A.	33	77	66	10	30	25	23	60	75	14	21	50	9	7
B.	77	33	66	10	30	25	23	60	75	14	21	50	9	7
C.	77	33	75	66	10	30	25	23	60	7	14	21	50	9
D.	77	33	7	66	10	30	25	23	60	75	14	21	50	9

5. Considere a árvore **binária, ordenada e balanceada AVL** representada na figura à direita.

Considere que se introduz na árvore um dos dois números seguintes: 91 ou 256. Qual das seguintes afirmações é **verdadeira**?



- A. A árvore permanece balanceada para qualquer daqueles dois números.  
 B. Apenas a introdução do número 91 impõe a necessidade de repor o balanceamento.  
 C. Apenas a introdução do número 256 impõe a necessidade de repor o balanceamento.  
 D. Qualquer dos dois números impõe a necessidade de repor o balanceamento.

6. Considere novamente a árvore **binária, ordenada e balanceada AVL** da questão anterior. Suponha que se introduz o número 72. Ignorando a necessidade de garantir e/ou verificar se a árvore permanece balanceada, indique qual das sequências abaixo seria produzida em varrimento in-fixado logo após a inserção daquele número e antes de se proceder à rotação para reposição do balanceamento, caso tal seja necessário.

- A. 23-49-56-72-67-73-86-98-107-113-140-158-174-202  
 B. 23-49-56-67-72-73-86-98-107-113-140-158-174-202  
 C. 23-49-56-67-72-73-86-98-113-107-140-158-174-202  
 D. 23-49-56-67-73-72-86-98-107-113-140-158-174-202

7. Seja uma tabela de dispersão de tamanho 100 com índices livres e em que as colisões se resolvem por procura linear. Nessa tabela pretende-se armazenar um conjunto de estruturas, utilizando como chave para a função de dispersão um dos campos da estrutura que contém números inteiros. Naturalmente que, sendo a tabela de tamanho 100, para cada número da chave a função de dispersão determina um índice entre 0 e 99, que corresponde ao resto da divisão inteira da chave por 100.

Suponha que os primeiros 10 registos introduzidos possuem as seguintes chaves: 25301; 32799; 56303; 14600; 28327; 43099; 66302; 43627; 17905; 50004.

Representando por '-' uma posição vazia, qual das tabelas abaixo representa correctamente o conteúdo das suas primeiras 8 posições?

A.	{43099; 25301; 14600; 56303; 66302; 17905; 50004; '-'}
B.	{14600; 25301; 66302; 56303; 50004; 17905; 43099; '-'}
C.	{14600; 25301; 43099; 56303; 66302; 17905; 50004; '-'}
D.	{14600; 25301; 66302; 56303; 50004; 17905; '-'; 28327}

8. Considere a seguinte recorrência

$$C_N = 3C_{N/2} + N^{\log_2(3)}$$

Por aplicação do Master Theorem, qual dos seguintes conjuntos representa a complexidade temporal associada com a recorrência dada?

A.	$\mathcal{O}(N^{\log_2(3)} \log(N))$	B.	$\mathcal{O}(N^{\log_2(3)})$	C.	$\mathcal{O}(N \log(N))$	D.	$\mathcal{O}(N \log^2(N))$
----	--------------------------------------	----	------------------------------	----	--------------------------	----	----------------------------

## PARTE II - Questões de Desenvolvimento

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

[5.00]

9. Considere um grafo ponderado não direccionado de 10 vértices e 23 arestas, cuja matriz de adjacências se apresenta abaixo.

$$\begin{bmatrix} 0 & 8 & 4 & 6 & \infty & \infty & \infty & \infty & \infty & \infty \\ 8 & 0 & 3 & \infty & 16 & 2 & \infty & 13 & \infty & \infty \\ 4 & 3 & 0 & 7 & 10 & 9 & \infty & \infty & \infty & 22 \\ 6 & \infty & 7 & 0 & 5 & \infty & \infty & \infty & \infty & 17 \\ \infty & 16 & 10 & 5 & 0 & 1 & 14 & \infty & \infty & 21 \\ \infty & 2 & 9 & \infty & 1 & 0 & 15 & 16 & \infty & \infty \\ \infty & \infty & \infty & \infty & 14 & 15 & 0 & 12 & 18 & 20 \\ \infty & 13 & \infty & \infty & \infty & 16 & 12 & 0 & 19 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 18 & 19 & 0 & 11 \\ \infty & \infty & 22 & 17 & 21 & \infty & 20 & \infty & 11 & 0 \end{bmatrix}$$

[3.00]

- a) Assumindo que a numeração dos vértices começa em zero, tome o vértice de índice 0 como ponto de partida para determinar a Árvore de Mínima de Suporte ("MST"), usando o algoritmo de Prim. Indique justificadamente todos os passos que executar e descreva em detalhe os cálculos realizados.

[0.75]

- b) A partir dos cálculos da alínea anterior, trace a "MST" que obteve.

[0.75]

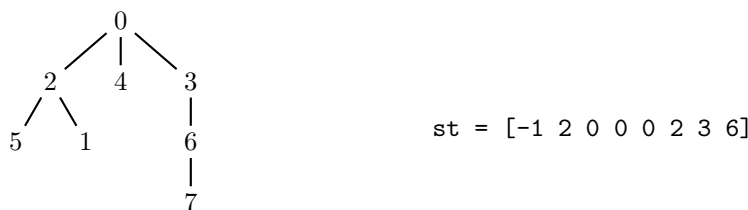
- c) Qual o custo da "MST" que obteve?

[0.50]

- d) Suponha que se adiciona uma aresta entre os vértices 0 e 4. Assumindo que os pesos das arestas são números inteiros, indique qual o valor mínimo do peso desta aresta para que o novo grafo continue a ter a mesma MST que o grafo original. A sua justificação não deverá necessitar calcular a MST para o novo grafo.

[4.00]

10. Considere que para uma rede foi aplicado o algoritmo de Dijkstra tomando todos os  $V$  vértices como fonte. A aplicação do algoritmo para cada vértice fonte,  $v$ , produz um vector  $\mathbf{st}$  de dimensão  $V$  que representa a SPT desse grafo para esse vértice. Assim, existem ao todo  $V$  vectores com aquela informação, que pode assumir estarem compactados numa matriz. Isto é, cada linha contém o vector  $\mathbf{st}$  que está associado ao vértice indexado pelo número dessa linha. Em baixo apresenta-se um exemplo de uma SPT com o vértice 0 como fonte, juntamente com o vector  $\mathbf{st}$  que lhe está associado.



Pretende-se desenvolver uma solução eficiente para o cálculo do caminho mais curto entre dois vértices dados. Assuma que a matriz de inteiros com a informação de caminhos é fornecida. Dado que cada caminho pode ter um número variável de vértices, deverá escolher a estrutura de dados adequada a essa indefinição para registar o caminho.

A função a desenvolver tem a seguinte assinatura:

```
caminho * produz_caminho(int ** M_st, int fonte, int destino);
```

- [0.50] a) Proponha justificadamente uma estrutura de dados para armazenar o caminho. Como a assinatura da função indica, o tipo de variável por si definido terá de se chamar `caminho`. Justifique a sua escolha.
- [0.75] b) Trace o fluxograma do algoritmo a desenvolver. Forneça uma descrição mínima de cada um dos blocos, descrevendo o algoritmo que vai implementar.
- [2.00] c) Escreva a função em C que concretize o fluxograma por si traçado e cumpra a função requerida.
- [0.75] d) Indique qual a complexidade temporal e de memória da função que produz o caminho mais curto entre dois vértices a partir de um vector `st` de dimensão  $V$ .
- [3.00] 11. Suponha que pretende armazenar um vasto conjunto de  $N$  números inteiros a que pretende aceder múltiplas vezes durante um longo período. Para resolver o problema é-lhe proposta uma solução suportada numa implementação de uma tabela de dispersão de dimensão  $M$ , com  $M < N$ , que se baseia no **método de dispersão por árvore ordenada**. Isto é, cada índice da tabela possui um ponteiro para uma árvore binária ordenada. Assuma que a função de dispersão garante uma distribuição aproximadamente uniforme por todas as posições da tabela.
- [1.25] a) Discuta justificadamente qual a complexidade temporal da operação de procura. A sua discussão deverá incluir a situação de procura com sucesso e com insucesso, assim como deverá explicitar a análise de caso médio e pior caso.
- [1.25] b) Assuma agora que a árvore ordenada é balanceada AVL. Repita a discussão de complexidade temporal da alínea anterior, tendo em conta aquela característica.
- [0.50] c) Em face da análise que produziu nas duas alíneas anteriores e no que à procura diz respeito, discuta possíveis vantagens e/ou desvantagens destas soluções relativamente ao **método de dispersão por lista simples**.
- [2.00] 12. Existem duas estratégias principais de desenvolvimento de algoritmos: *divide-and-conquer* e programação dinâmica.
- [1.00] a) Caracterize cada uma delas, salientando as suas propriedades distintivas e situações de aplicabilidade. Se entender útil, dê exemplos de problemas onde cada uma das duas técnicas é aplicável com vantagem de eficiência computacional e dê também exemplos do contrário.
- [1.00] b) No caso específico da programação dinâmica existem duas abordagens alternativas: ascendente e descendente. Descreva as diferenças e semelhanças entre estas duas alternativas. Também será útil que forneça um exemplo que ilustre como é que estas duas abordagens alternativas diferem entre si.