



1º Exame, 21 de Junho de 2018, 18h30m 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. No algoritmo de conectividade com 4 nós e inicialização $id[i] = i$, a tabela $[2 \ 0 \ 2 \ 1]$ representa a conectividade de um grafo, tendo sido obtida por um algoritmo do tipo união rápida (união rápida simples ou união rápida equilibrada ou união rápida equilibrada com compressão de caminho). Se tivesse sido utilizado um algoritmo de procura rápida, qual das seguintes tabelas poderia ser obtida?

A. $[2 \ 0 \ 3 \ 0]$	B. $[0 \ 0 \ 3 \ 3]$	C. $[0 \ 3 \ 0 \ 2]$	D. $[1 \ 1 \ 1 \ 1]$
----------------------	----------------------	----------------------	----------------------

2. Seja a tabela abaixo (primeira linha) a qual se aplicaram as primeiras quatro iterações de um dos algoritmos de ordenação estudados durante o semestre, resultando na tabela presente na segunda linha.

5	6	11	9	8	16	11	6	18
5	6	11	6	8	16	11	9	18

Qual será a tabela resultante na quinta iteração?

A.	<table><tr><td>5</td><td>6</td><td>8</td><td>9</td><td>11</td><td>16</td><td>11</td><td>6</td><td>18</td></tr></table>	5	6	8	9	11	16	11	6	18	B.	<table><tr><td>5</td><td>6</td><td>11</td><td>9</td><td>6</td><td>16</td><td>11</td><td>8</td><td>18</td></tr></table>	5	6	11	9	6	16	11	8	18
5	6	8	9	11	16	11	6	18													
5	6	11	9	6	16	11	8	18													
C.	<table><tr><td>5</td><td>6</td><td>11</td><td>6</td><td>8</td><td>16</td><td>11</td><td>9</td><td>18</td></tr></table>	5	6	11	6	8	16	11	9	18	D.	<table><tr><td>5</td><td>6</td><td>8</td><td>9</td><td>11</td><td>11</td><td>16</td><td>6</td><td>18</td></tr></table>	5	6	8	9	11	11	16	6	18
5	6	11	6	8	16	11	9	18													
5	6	8	9	11	11	16	6	18													

3. Dada a tabela de caracteres abaixo, suponha que essa tabela é processada da primeira posição em diante, do seguinte modo. Existe uma pilha, para a qual os elementos desta tabela são enviados sequencialmente por cada vez que se executa a operação **push** – p. Cada operação **pop** – P remove o elemento da pilha e imprime para terminal.

2	e	a	0	e	2	d	8	1
---	---	---	---	---	---	---	---	---

Qual das sequências de operações de manipulação da pilha, produz a palavra *aed2018*?

A. pppPppPppPppPPpPP	B. pppPppPppPPPPpPpP
C. pppPppPppPPPPpPpP	D. pppPppPppPPPPpPpP

4. Após se ter aplicado o algoritmo de Prim a um grafo de 10 vértices, numerados de 0 a 9, obteve-se o vector **st**, representando a árvore resultante, que abaixo se apresenta:

9	5	7	6	-1	4	4	5	5	6
---	---	---	---	----	---	---	---	---	---

Qual das seguintes afirmações é falsa?

- A. O grafo é ligado.
- B. Não existe a aresta (5, 8).
- C. A sequência de vértices 3-6-4-5-7-2 constitui um caminho válido neste grafo.
- D. Se existir, a aresta (3, 9) não possui menor custo que as arestas (3, 6) e (6, 9).

5. Uma função recursiva pode ter a sua complexidade temporal descrita da seguinte forma: $C_N = 2C_{N/2} + f(N)$, em que $f(N)$ é uma função de complexidade $\mathcal{O}(N \log^2 N)$. Escolha qual a afirmação verdadeira.

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

6. Considere a seguinte função, cujo o argumento representa um ponteiro para o nó de uma árvore binária:

```
int func(link *h)
{
    if (h == NULL) return 0;
    if (h->l != NULL || h->r != NULL) return 1 + func(h->l) + func(h->r);
}
```

Diga qual a funcionalidade desta função:

- A. Cálculo do número de nós da árvore.
- B. Cálculo do número de folhas da árvore.
- C. Cálculo da altura da árvore.
- D. Cálculo do número de nós que não são folhas.

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. Numa árvore ordenada de pesquisa binária, as operações de rotação preservam sempre a ordenação.
8. No problema da conectividade com algoritmo quick union ponderado, se $id[i] \neq id[j]$ então os elementos i e j não estão conectados.
9. O algoritmo de Kruskal é preferível ao algoritmo de Prim para a determinação de árvores mínimas de suporte (MST) quando os grafos são densos.
10. A implementação de uma pilha através de uma tabela é, em geral, mais eficiente do que a implementação por lista, seja em termos de memória, seja em termos temporais.
11. No algoritmo de ordenação *shell-sort*, uma qualquer tabela que esteja $2h$ -ordenada também está h -ordenada, para qualquer $h > 1$.
12. Para uma tabela, H , representando um acervo binário, o elemento $H[9]$ possui sempre prioridade menor ou igual à do elemento $H[3]$.
13. Aplicar procura em largura (BFS) a um qualquer grafo produz sempre o caminho mais curto entre um qualquer vértice, passado como fonte, e outro, passado como destino, independentemente dos pesos associados às arestas.

14. Uma tabela de dispersão com 100 posições, em que a função de dispersão é o resto da divisão inteira por 100, consegue indexar no máximo 100 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.

- [4.5] 15. Considere uma operação em tabelas que calcula, para cada elemento da tabela, a média aritmética dos seus M vizinhos. Considere M ímpar, pelo que o valor da média na posição i usa os valores da tabela originais na gama de índices $[i - M/2, i + M/2]$. Se os extremos deste intervalo saem dos limites da tabela original o valor é zero.

Seja $x[]$ a tabela original e $y[]$ a tabela com os valores correspondentes da média, ambas de dimensão N .

Uma possível implementação é a seguinte (por simplicidade não se apresenta código de tratamento de erros):

```
void mediatgab(float *x, float *y, int N, int M)
/*assume x e y alocados de dimensão N, M ímpar*/
{
    int n = 0, m = 0, halfM = M/2;
    for (n = 0; n < N; n++) {
        y[n] = 0;
        if(n < halfM || n >= N-half)
            continue;
        for (m = halfM; m <= halfM; m++)
            y[n] += x[n+m];
        y[n] /= M;
    }
}
```

- [1.0] a) Assuma que $N = 100 * M$. Diga, em função de N , qual a complexidade deste algoritmo. Justifique adequadamente o seu raciocínio e conclusões.
- [1.0] b) Proponha um algoritmo que produza o mesmo resultado com complexidade $\mathcal{O}(N)$.
- [2.5] c) Escreva a função em linguagem C que implementa o algoritmo por si proposto.

Responda em folhas separadas. Não se esqueça de mudar de folha

- [4.5] 16. Considere um grafo ponderado não direccionado com 12 vértices, identificados de 0 a 11, representado pela matriz de adjacências indicada abaixo. A ausência de valor para um par linha e coluna significa ausência de aresta entre esses dois vértices, i.e., custo ∞ .

	0	1	2	3	4	5	6	7	8	9	10	11
0	0			7		9	11		5		8	6
1		0	3		5	7		2				4
2		3	0			6		4		2		
3	7			0			1				4	
4		5			0			3	7			1
5	9	7	6			0	2			1		4
6	11			1		2	0			5		
7		2	4		3			0				
8	5				7				0		3	8
9			2			1	5			0		
10	8			4					3		0	
11	6	4			1	4			8			0

- [3.0] a) Tomando o vértice 7 como ponto de partida determine a árvore de caminhos mais curtos de fonte única.
- [0.5] b) Trace a SPT que resultou dos cálculos da alínea anterior.
- [1.0] c) Caso a árvore que calculou anteriormente o permita, diga qual o caminho mais curto entre o vértice 9 e o vértice 3. Indique o custo desse caminho.
- Se a árvore que calculou anteriormente não lhe permitir aquela determinação, explique porquê.

Responda em folhas separadas. Não se esqueça de mudar de folha

- [2.5] 17. As palavras de língua inglesa *fresher* e *refresh* são anagramas. Ou seja, são produzidas pelos mesmos caracteres. Outro exemplo são as palavras de língua portuguesa *saca* e *casa*.

Suponha que se pretende resolver o seguinte problema. Dado um qualquer dicionário de uma qualquer língua, em que as palavras são constituídas pelos caracteres latinos, pretende-se produzir um ficheiro de saída em que cada uma das suas linhas contém todos os anagramas existentes nesse dicionário para algum conjunto de caracteres.

Por exemplo, um troço do ficheiro de saída poderia ser:

```
...
kinship pinkish
enlist inlets listen silent
boaster boaters borates
...
```

Para simplificar o problema, assuma que os dicionários não possuem palavras hifenadas nem palavras com caracteres acentuados nem cedilhados, assim como todas são constituídas por caracteres minúsculos.

- [1.5] a) Sem escrever código proponha um par algoritmo & estrutura de dados que permita resolver este problema da forma mais eficiente possível. A sua resposta deverá ser clara, sem ambiguidades e suficientemente detalhada. Se entender útil, pode desenhar um fluxograma que permita compreender todo o procedimento. O procedimento deverá receber um ficheiro como argumento que contém todas as palavras do dicionário a analisar. A sua descrição deverá contemplar aspectos essenciais relativos ao armazenamento desse dicionário em memória, como procederá para identificar todos os anagramas existentes e imprimi-los no ficheiro de saída.
- [1.0] b) Assumindo que o dicionário passado como argumento possui P palavras e a maior delas possui tamanho k , indique qual a complexidade do procedimento por si proposto. A discussão de complexidade, deverá ser devidamente fundamentada e deverá contemplar a memória necessária bem como o tempo gasto, como função dos dois parâmetros que caracterizam o problema, P e k .