



1º Exame, 17 de Janeiro de 2020, 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. Qual a complexidade da função abaixo?

```
void f(int N)
{
    if (N == 0)
        return;
    for (int j = 0; j <= N; ++j)
        f(j);
}
```

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

2. No contexto do problema da conectividade quer-se saber qual o menor número de uniões que devem ser realizadas no algoritmo da união rápida ponderada para que a altura da árvore do maior conjunto seja 5.

- A. 4 B. 15 C. 31 D. 127

3. Sobre o código que se apresenta abaixo, pode-se dizer que...

```
float **m = (float **) malloc(10 * sizeof (float *));
for (int i=0; i++; i < 10) {m[i] = malloc(20 * sizeof (float));}
/* corpo principal do programa */
for (int i=0; i++; i < 20) {free(m[i]);}
free(m);
```

- A. ... cria uma matriz com 10 linhas e 20 colunas, que liberta corretamente.
B. ... cria uma matriz com 10 linhas e 20 colunas, que liberta incorretamente.
C. ... cria uma matriz com 20 linhas e 10 colunas, que liberta corretamente.
D. ... cria uma matriz com 20 linhas e 10 colunas, que liberta incorretamente.

4. Suponha que tem uma tabela de dispersão com resolução de colisões por separação em listas com inserção no final. Assuma que a função de dispersão é $h(k) = k \bmod 11$ e que insere na tabela, por ordem, os seguintes elementos: 30, 20, 56, 75, 31, 19.

Qual é o maior número de comparações entre chaves que pode vir a fazer numa procura, nesta tabela:

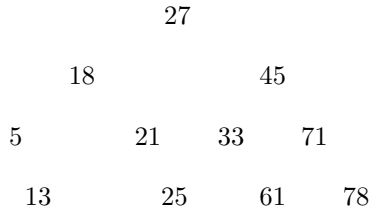
- A. 1 B. 2 C. 3 D. 4

5. Para a tabela de inteiros abaixo aplicou-se o algoritmo *shellsort* em que o primeiro valor de h usado foi 3. Diga qual das hipóteses é a tabela resultante após se concluírem todas as iterações para $h = 3$.

| 9 | 7 | 6 | 2 | 8 | 5 | 3 | 1 |

A.	2 7 5 3 1 6 9 8	B.	2 1 6 3 7 5 9 8
C.	2 1 5 3 7 6 9 8	D.	1 2 3 5 6 7 8 9

6. Considere a árvore **binária, ordenada e balanceada AVL** que se apresenta abaixo à esquerda.



Suponha que se introduzem os seguintes números: 1, 15, 19, 30. Qual das sequências de introdução é tal que a árvore perde balanceamento? Isto é, qual a sequência que requer a realização de, pelo menos, uma rotação?

A.	1, 15, 30, 19	B.	30, 1, 19, 15	C.	30, 15, 19, 1	D.	19, 1, 30, 15
----	---------------	----	---------------	----	---------------	----	---------------

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

Preencha as respostas na tabela (usando apenas letras maiúsculas – V(erdadeira) ou F(alsa)). Cada questão de escolha binária vale 0.50 valores. As questões não respondidas ou erradas são cotadas com 0 valores.

Questão	7	8	9	10	11	12	13	14
Resposta								

- O varrimento in-fixado só produz output ordenado se a árvore ordenada também for balanceada AVL.
- A técnica de projecto designada como programação dinâmica, na versão ascendente, admite normalmente implementações não recursivas.
- No algoritmo de ordenação *quicksort* a seleção de pivot através da mediana de três garante sempre que a partição executada produz duas tabelas de tamanhos semelhantes, embora não necessariamente iguais.
- A DFS encontra sempre solução em problemas definidos em grafos finitos ligados, *i.e.*, grafos ligados com um número finito de vértices.
- Armazenar dados em tabelas, em vez de listas, é preferível se as operações mais frequentes do algoritmo forem acesso e modificação.
- Dada uma lista de N elementos, é teoricamente possível determinar se todos os elementos são distintos em tempo $\mathcal{O}(N)$.
- Um acervo é uma representação em tabela de uma árvore binária balanceada satisfazendo a condição de acervo. Se a prioridade for tanto maior quando menor for o número, encontrar e remover o menor elemento de um acervo tem a mesma complexidade que encontrar e remover o menor elemento de uma árvore binária ordenada e balanceada.
- A recorrência $C_N = C_{2N} + N$ descreve um programa recursivo que não termina.

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] 15. Os números naturais estritamente menores que 10 que são múltiplos de 3 ou de 5 são: 3, 5, 6, e 9. A soma destes múltiplos é $S_{(3,5)}(10) = 23$. Pretende-se escrever uma função que calcule a soma dos múltiplos de 3 e de 5 que sejam estritamente menores que um dado inteiro. Por exemplo, para $N = 1000$ quanto valerá $S_{(3,5)}(1000)$?
- [2.0] a) Proponha um algoritmo para resolver este problema, fazendo uso de uma descrição por texto e/ou socorrendo-se de um fluxograma suficientemente claro e devidamente explicado. Seja preciso e conciso na sua descrição.
- [1.5] b) Escreva a função `int sum_3_5(int n)` que implementa o algoritmo por si proposto, não se esquecendo de a comentar devidamente, para que seja claro o que cada troço do seu código está a (ou pretende) fazer.
- [0.5] c) De forma devida e adequadamente detalhada e justificada determine a complexidade temporal do seu algoritmo, como função do parâmetro de entrada.

Nota: Para obter a cotação completa deste exercício não basta que a função esteja correcta. O algoritmo proposto e sua implementação tem de ser o mais eficiente possível. Não se esqueça que, por exemplo, $S_{(3,5)}(3) = 0$.

- [4.5] 16. Para um grafo não direccionado de 14 vértices, numerados de 0 a 13, aplicou-se o algoritmo de Prim e o algoritmo de Dijkstra, tomando em ambos os casos o mesmo vértice de partida. Após a conclusão do algoritmo de Prim obtiveram-se as seguintes tabelas indexadas pelos vértices:

$$\begin{aligned}st_P &= [1 | 8 | 11 | 13 | 9 | 1 | 13 | 2 | 12 | 13 | 2 | -1 | 2 | 1] \\wt_P &= [2 | 2 | 4 | 3 | 4 | 3 | 3 | 3 | 1 | 1 | 1 | 0 | 2 | 1]\end{aligned}$$

Após a aplicação do algoritmo de Dijkstra obtiveram-se as tabelas abaixo:

$$\begin{aligned}st_D &= [1 | 10 | 11 | 1 | 12 | 10 | 13 | 2 | 12 | 13 | 11 | -1 | 2 | 1] \\wt_D &= [11 | 9 | 4 | 13 | 11 | 11 | 13 | 7 | 7 | 11 | 4 | 0 | 6 | 10]\end{aligned}$$

- [0.50] a) Trace a árvore representada pelo par st_P e wt_P . Justifique o seu traçado.
- [0.25] b) Usando a árvore anterior produza o caminho entre os vértices 10 e 13 e indique o seu custo, ou justifique porque não se pode obter.
- [0.50] c) Trace a árvore representada pelo par st_D e wt_D . Justifique o seu traçado.
- [0.25] d) Usando a árvore anterior produza o caminho entre os vértices 10 e 13 e indique o seu custo, ou justifique porque não se pode obter.
- [1.00] e) Indique quantas e quais as arestas de peso unitário que são usadas em cada uma das duas árvores. Justifique os seus cálculos e conclusões.
- [1.00] f) Caso os números que obteve na alínea anterior sejam diferentes, havendo uma das árvores em que são usadas menos arestas de peso 1 que na outra, explique a razão de tal assim ser.
- [1.00] g) O grafo possui uma aresta entre o vértice 0 e o vértice 12. No entanto, essa aresta não está presente em nenhuma das duas árvores. Indique o custo mínimo que essa aresta deverá possuir que justifique esta ausência.

- [3.0] 17. Uma tabela de dispersão é uma estrutura de dados muito utilizada em problemas em que frequentemente se adicionam dados novos, se procuram dados ou se removem dados já armazenados.

Uma forma de implementar uma tabela de dispersão é o método dos índices livres. Na sua forma mais simples, dado um elemento, a chave do elemento é passada pela função de dispersão retornando um índice. Se o objetivo é introduzir o elemento na tabela, acede-se a esse índice e verifica-se se essa posição está vazia. Se estiver, o elemento é introduzido aí e o processo termina. Se não estiver vazio, o algoritmo procura em sequência a primeira posição vazia e introduz aí o elemento. Qualquer procura posterior pelo elemento procede da mesma maneira: encontra o elemento na posição dada pela função de dispersão; ou procura-o sequencialmente na tabela a partir dessa posição (o conjunto de números localizados em posições diferentes dessa mas contíguos denomina-se um cluster). Se o elemento é encontrado retorna-se sucesso, se se atinge uma casa vazia sem o encontrar, conclui-se que o elemento não está na tabela.

Suponha uma tabela de dispersão de dimensão 31, com dispersão por índices livres, com uma função de dispersão dada por $f(x) = x \bmod 31$.

- [0.5] a) Introduza por ordem na tabela os números 7, 4, 9, 35, 8, 38 e 69. Indique o estado da tabela neste momento.
- [0.5] b) Se a tabela tiver agora M elementos e nela forem introduzidos N elementos distintos ($N < M$), admitindo que podemos querer introduzir qualquer número com igual probabilidade, qual a complexidade média da inserção em termos de N ? No melhor e no pior caso como é que essa complexidade se altera, se é que se altera?

Nesta alínea e nas seguintes vamos lidar com o problema da remoção de um elemento de uma tabela de dispersão.

- [0.5] c) Suponha que se pretende remover um dado elemento de uma tabela de dispersão por índices livres com M posições. Dada a chave do elemento a função de dispersão permite procurá-lo da forma anteriormente descrita. Assuma que ele é encontrado. Explique porque é que simplesmente remover o elemento da tabela após ele ter sido encontrado pode não permitir de futuro encontrar outros números na tabela.
Sugestão: Utilize o exemplo de a) para tentar perceber qual é o problema se por exemplo quiser remover o elemento 38.
- [0.5] d) Uma possibilidade melhor talvez seja percorrer o cluster até ele terminar e determinar qual o último elemento nesse cluster para o qual a função de dispersão a posição de início da procura (no caso da alínea a) seria o elemento 69 e colocá-lo na posição de onde se removeu o elemento. No exemplo da alínea a), se se pretender retirar o item 7 ele seria removido, mas o elemento 69 seria colocado na posição 7 da tabela. Infelizmente, apesar de esta estratégia permitir manter o cluster junto, ela não funciona. Explique porquê.
Sugestão: Utilize o exemplo da alínea a), introduza cuidadosamente um novo elemento e depois retire o elemento 7.
- [1.0] e) Proponha uma solução alternativa que garanta a remoção do elemento e não cause os problemas indicados. Explique como é que fazendo a remoção dessa forma a tabela continua a funcionar para inserções, procuras ou remoções seguintes.

Sugestão: Assuma que além dos números pode introduzir na tabela identificadores especiais por si definidos, como **VAZIO** ou outro qualquer que você defina com um significado apropriado e que as funções de manipulação de dados compreendem.