

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 o problema da conectividade, com $id = [0, 1, 9, 4, 9, 6, 9, 7, 0, 9]$, e a seguinte função em C que realiza procura.

```
int isConnected (int i, int j) {  
    while (i != id[i]) i = id[i];  
    while (j != id[j]) j = id[j];  
    return (i == j);  
}
```

Qual o par correspondente a $(isConnected(5,8), isConnected(2,3))$?

A. (0, 0)	B. (0, 1)	C. (1, 0)	D. (1, 1)
-----------	-----------	-----------	-----------

2. Considere operações realizadas sobre conjuntos de itens. O resultado da operação é escrito sobre um dos operandos. Utilizando o critério de eficiência temporal, i.e. mais rápido na execução, faça corresponder o par de operações (União, Intersecção) com um dos seguintes pares de estruturas de dados:

A. (Lista, Tabela)	B. (Tabela, Lista)	C. (Tabela, Tabela)	D. (Lista, Lista)
--------------------	--------------------	---------------------	-------------------

3. Considere a seguinte função de reposição da ordenação de um acervo ("heap") quando a prioridade de um nó é aumentada:

```
1: void FixUp(Item Heap[], int Idx) {  
2:   if (Idx == 0) return;  
3:   while (Idx > 0 && lessPri(Heap[(Idx-1)/2], Heap[Idx])) {  
4:     exch(&Heap[Idx], &Heap[(Idx-1)/2]);  
5:     Idx = (Idx+1)*2;  
6:   }  
7: }
```

A função `lessPri(Item x, Item y)` devolve 1 se a prioridade de `x` for menor do que a de `y` e devolve 0 em caso contrário. A função `exch(Item *x, Item *y)` troca os valores das variáveis apontadas por `x` e `y`. Em que linha existe um erro?

Nota: Entre outras coisas, tome particular atenção aos protótipos definidos para as funções e que só uma das linhas poderá estar errada.

A. Na linha 2	B. Na linha 3	C. Na linha 4	D. Na linha 5
---------------	---------------	---------------	---------------

4. Suponha que numa tabela de dispersão ("hash table") são introduzidos 5 números pela seguinte ordem: {401, 44, 766, 802, 560}. Os 4 primeiros elementos da tabela de dispersão obtida são os seguintes [560, 401, 766, 802, ...]. Note-se que nada é dito sobre o número de elementos da tabela de dispersão (pode ter 4 ou mais elementos). As colisões são resolvidas por procura linear. Qual a função de dispersão usada?

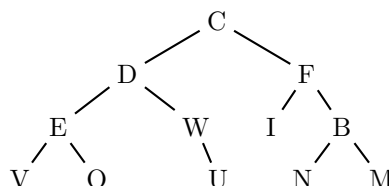
- A. Resto da divisão inteira por 10
- B. Soma dos dois algarismos mais à direita
- C. Resto da divisão inteira por 5
- D. Um mais a soma dos dois algarismos mais à direita

5. Considere uma árvore binária ordenada, não necessariamente balanceada. Complete a frase seguinte com a opção **verdadeira**. Sendo N o número de nós, a pesquisa numa árvore binária ordenada tem uma complexidade associada...

- | | |
|--------------------------------------|--|
| A. ... que é sempre $\mathcal{O}(N)$ | B. ... que é sempre $\mathcal{O}(\lg N)$ |
| C. ... que pode ser $\mathcal{O}(N)$ | D. ... que pode ser $\mathcal{O}(N^2)$ |

6. Considere o troço de código abaixo à esquerda e a árvore **binária** representada à direita.

```
void traverse(tree *h, void (*visit)(link))
{
    if (h == NULL) return;
    traverse(h->left, visit);
    (*visit)(h);
    traverse(h->right, visit);
}
```



A função disponibilizada implementa uma estratégia de varrimento onde a função `visit()` imprime a letra correspondente a cada um dos nós. Suponha que utiliza a função disponibilizada aplicada à árvore da figura. Qual das afirmações é verdadeira?

- A. O varrimento é Pré-Fixado com sequência de saída C D E V O W U F I B N M.
- B. O varrimento é In-Fixado com sequência de saída V E O D W U C I F N B M.
- C. O varrimento é In-Fixado com sequência de saída C D E V O W U F I B N M.
- D. O varrimento é Pós-Fixado com sequência de saída V O E U W D I N M B F C.

7. Usando a notação assintótica estudada e dada a expressão de recorrência $C_N = C_{N-1} + N$, indique qual das expressões seguintes é **verdadeira**:

- | | |
|-----------------------------------|---|
| A. $C_N \in \mathcal{O}(\lg_3 N)$ | B. $C_N \in \mathcal{O}(\lg 2^{(N^2/2)})$ |
| C. $C_N \in \mathcal{O}(N)$ | D. $C_N \in \mathcal{O}(\sqrt[3]{\lg N})$ |

8. Considere a seguinte tabela (primeira linha) sobre a qual são listados alguns passos executados por um algoritmo de ordenação (restantes linhas). Qual é o algoritmo usado?

3	3	9	7	1	0	9	2	3	7	3	5
1	3	9	7	3	0	9	2	3	7	3	5
1	0	9	7	3	3	9	2	3	7	3	5
1	0	3	7	3	3	9	2	3	7	9	5
1	0	3	2	3	3	9	5	3	7	9	7
1	0	3	2	3	3	3	5	9	7	9	7
1	0	3	2	3	3	3	5	9	7	9	7

- A. Inserção
- B. *Shellsort* ($h=4, 3, 1$)
- C. *Shellsort* ($h=4, 2, 1$)
- D. *Quicksort*

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.

[6.0]

9. Considere que no seu programa para gerar palavras cruzadas necessita determinar uma lista de palavras que satisfazem um conjunto de restrições. Suponha que existem K restrições e que possui uma tabela de K listas, tal que na lista k , $0 \leq k < K$, estão os índices de todas as palavras que satisfazem a restrição $k + 1$. Essa tabela de listas será produzida sempre que se pretende completar uma palavra no tabuleiro.

O seu objectivo é produzir uma lista que contenha apenas os índices que são comuns a todas as K listas, produzindo assim todos os índices candidatos à posição a completar.

Suponha que criou o tipo abstracto `lista` de acordo com a definição abaixo e que possui uma implementação das funções necessárias para criar, manipular e apagar elementos de variáveis do tipo `lista`.

```
typedef struct _lista{
    int indice_palavra;
    _lista * proxima;
} lista;
```

Assuma ainda que existem ao todo N palavras, indexadas numa tabela para $0 \leq n < N$ e que cada uma das listas possui M_k elementos, sendo este valor desconhecido à partida, enquanto que o valor total de palavras no dicionário é conhecido. Pode também assumir que cada uma das listas não está ordenada.

[2.0]

- a) Escreva o código de uma função **temporalmente eficiente** em C que receba duas listas de índices como argumento e retorne a lista contendo a intersecção das duas. Note que se uma lista for vazia a intersecção das duas listas será naturalmente vazia. A função que se pretende que escreva tem a seguinte assinatura:

```
lista * interseccao2(lista *lista1, lista * lista2);
```

[1.0]

- b) Determine justificadamente a complexidade temporal da função que escreveu em a), como função de N e/ou do tamanho de cada uma das duas listas, M_1 e M_2 . Pode assumir que $M_i \ll N$.

[2.0]

- c) Fazendo uso da função que escreveu em a) escreva agora uma função **eficiente** que produza a intersecção de um número arbitrário de listas, com a seguinte assinatura:

```
lista * interseccaok(lista **tabela_listas, int k);
```

A variável `tabela_listas` contém as listas a intersectar e a variável `k` indica a dimensão da tabela. Mesmo que não tenha escrito a função `interseccao2`, assuma que ela está disponível para esta alínea, de acordo com a assinatura especificada.

NOTA: Se `k` for igual a um, o resultado é a única lista da tabela.

[1.0]

- d) Como função de N , K e/ou M_k , determine a complexidade temporal da função `interseccaok`. Pode assumir, por conveniência de análise, que o tamanho médio das listas é M . Caso não tenha resolvido a alínea a) e/ou b) assuma que a complexidade da função `interseccao2` é alguma função $f(N, M_1, M_2)$ ou $f(N, M)$, com M identificando o valor médio do tamanho de cada lista ou o valor da maior lista, como achar conveniente. Indique qual o caso que considerou.

[4.0]

10. Considere que pretende determinar o valor da seguinte função: $f(p, q) = pf(p-1, q) + qf(p, q-1)$, definida para valores inteiros não negativos de p e q , com $f(0, q) = 0$ para todo o valor de $q \neq 0$, $f(p, 0) = 1$ para todo o $p \neq 0$, e $f(0, 0) = 2$.

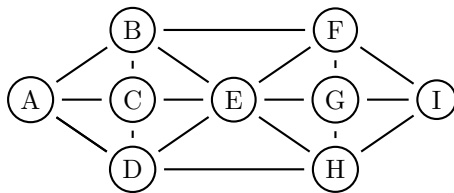
[2.0] a) Escreva o código de uma função **temporalmente eficiente** em C com a seguinte assinatura:

```
int f_p_q(int p, int q);
```

[1.0] b) Considerando as várias técnicas de desenvolvimento de algoritmos – "divide-and-conquer", programação dinâmica ascendente e programação dinâmica descendente –, qual delas utilizou na alínea a)? Justifique a opção e a resposta.

[1.0] c) Indique justificadamente quais os requisitos de memória da função que implementou.

[4.0] 11. Considere o grafo indicado em baixo e assuma que o mesmo não é direccionado mas é ponderado, como se indica do lado direito do grafo.



$E \leftrightarrow F$, $E \leftrightarrow D$ 1
 $A \leftrightarrow B$, $C \leftrightarrow D$, $B \leftrightarrow F$, $F \leftrightarrow G$, $H \leftrightarrow I$, $D \leftrightarrow H$ 2
 $A \leftrightarrow C$, $G \leftrightarrow I$, $B \leftrightarrow C$, $G \leftrightarrow H$ 3
 $B \leftrightarrow E$, $E \leftrightarrow H$ 4
 $C \leftrightarrow E$, $E \leftrightarrow G$ 5
 $A \leftrightarrow D$, $F \leftrightarrow I$ 6

[2.5] a) Determine a árvore de caminhos mais curtos (SPT) tomando o vértice **A** como fonte. Apresente os seus cálculos de forma clara, detalhada e completa para cada iteração do algoritmo. Por exemplo, mas sem se restringir a estes aspectos, identifique a franja da procura e pesos, assim como deverá indicar por que ordem entra cada vértice na árvore e o estado da franja em cada momento.

[0.75] b) Indique quais os vértices que fazem parte do caminho mais curto entre o vértice **A** e o vértice **G**, bem como qual o seu valor de custo.

[0.75] c) Diga se a SPT que calculou lhe permite ou não dizer qual o caminho mais curto entre **G** e **H**. Porquê?