



1º Exame, 06 Junho 2015, 08:00h 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 uma grelha de 4×4 , com as células numeradas de 0 a 15, na qual se descreve um labirinto. Duas células encontram-se ligadas se não existir uma parede entre elas. O conjunto de células ligadas de um labirinto pode ser representado por um problema de conectividade. Supondo que o ponto de saída do labirinto encontra-se na célula 15, e que o conjunto de células ligadas foi processado pelo algoritmo **QuickUnion** tendo resultado na seguinte tabela:

4	5	3	6	8	2	7	11	9	9	10	15	13	14	10	15
---	---	---	---	---	---	---	----	---	---	----	----	----	----	----	----

Indique das hipóteses seguintes qual a célula inicial que permite atingir a saída do labirinto representado na tabela anterior.

A.	0.	B.	12.	C.	3.	D.	10.
----	----	----	-----	----	----	----	-----

2. Considere a inserção do seguinte sequência de números numa árvore ordenada balanceada AVL, inicialmente vazia.

$\{14, 17, 11, 7, 53, 4, 13, 12, 8\}$

Das hipóteses seguintes identifique qual o número total acumulado de rotações necessário para que a árvore continue ordenada e balanceada AVL no final e após cada inserção. Considere que uma rotação dupla conta como duas rotações para a contagem pretendida.

A.	3.	B.	4.	C.	5.	D.	7.
----	----	----	----	----	----	----	----

3. Se cada nó de uma árvore binária possuir zero ou dois filhos, com todos os níveis preenchidos excepto possivelmente o último, a altura dessa árvore binária é dada pela expressão:

A.	$n \log_2(n)$.	B.	$n \log_2(n + 1)$.	C.	$\log_2(n)$.	D.	$\log_2(n + 1)$.
----	-----------------	----	---------------------	----	---------------	----	-------------------

4. Considere o código seguinte que define uma função C que determina de forma recursiva o valor máximo num vector de inteiros.

```

int findmax(int * array, int size)
{
    int max;

    if (size == 1) return array[0];
    else {
        max = findmax(array, size-1);
        if (max >= array[size-1]) return max;
        else return array[size-1];
    }
}

```

Assumindo que o array tem dimensão n , das hipóteses seguintes identifique qual o par que corresponde à recorrência/solução que descreve a recursividade da função `findmax`.

- | | | | | | |
|----|---------------------|-----------------------|----|-------------------------|-------------------|
| A. | $c(n) = c(n-1) + 1$ | $c(n) = O(n)$. | B. | $c(n) = c(n-1) + n - 1$ | $c(n) = O(n)$. |
| C. | $c(n) = c(n/2) + 1$ | $c(n) = O(\log(n))$. | D. | $c(n) = c(n-1) + 1$ | $c(n) = O(n^2)$. |

5. Seja o algoritmo `QuickSort` aplicado à tabela seguinte com o objectivo de a ordenar crescentemente.

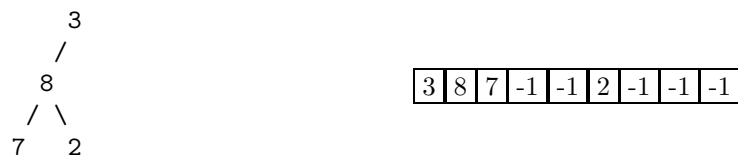
5	19	16	13	22	4	7	10
---	----	----	----	----	---	---	----

Pretende-se identificar a sequência de pivots utilizada pelo algoritmo. Das hipóteses seguintes, indique qual a sequência correcta de pivots para a tabela anterior.

- | | | | |
|----|-------------------------|----|-------------------------|
| A. | {10, 4, 5, 22, 13, 19}. | B. | {10, 5, 4, 13, 19, 22}. |
| C. | {10, 5, 4, 22, 19, 13}. | D. | {10, 4, 5, 13, 22, 19}. |

6. Considere a expressão matemática $((6 \times (4 + 5) - 8) / (2 + 3))$, a qual se pretende representar numa árvore binária, em que cada nó ou é um algarismo ou um símbolo aritmético $\{+, -, \times, /\}$. Suponha que para a representação da árvore se utiliza um vector de acordo com o formato utilizado no laboratório 3, ou seja, no vector existem números inteiros maiores ou iguais a zero, ou iguais a -1. Um número maior ou igual a zero significa que a árvore tem um nó com esse número. O número -1 significa que um nó pai não tem um filho.

A seguir, apresenta-se um exemplo de uma árvore e do vector que a representa:

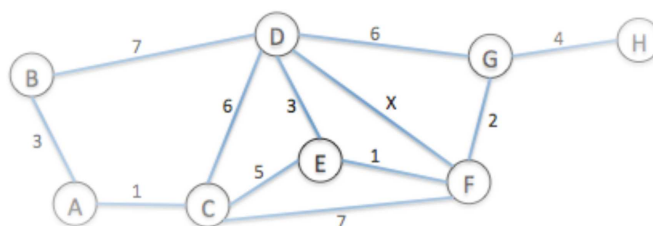


Pretende-se determinar que árvores devem ser definidas de forma a que o respectivo varrimento pré-fixado, in-fixado e pós-fixado resulta na expressão matemática anterior.

Das opções seguintes indique qual **não** permite obter a expressão pretendida.

- | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|
| A. | <table border="1"><tr><td>/</td><td>-</td><td>x</td><td>6</td><td>-1</td><td>-1</td><td>+</td><td>4</td><td>-1</td><td>-1</td><td>5</td><td>-1</td><td>-1</td><td>8</td><td>-1</td><td>-1</td><td>+</td><td>2</td><td>-1</td><td>-1</td><td>3</td><td>-1</td><td>-1</td></tr></table> | / | - | x | 6 | -1 | -1 | + | 4 | -1 | -1 | 5 | -1 | -1 | 8 | -1 | -1 | + | 2 | -1 | -1 | 3 | -1 | -1 | in-fixado. |
| / | - | x | 6 | -1 | -1 | + | 4 | -1 | -1 | 5 | -1 | -1 | 8 | -1 | -1 | + | 2 | -1 | -1 | 3 | -1 | -1 | | | |
| B. | <table border="1"><tr><td>3</td><td>8</td><td>5</td><td>6</td><td>-1</td><td>-1</td><td>+</td><td>x</td><td>-1</td><td>-1</td><td>4</td><td>-1</td><td>-1</td><td>-</td><td>-1</td><td>-1</td><td>+</td><td>/</td><td>-1</td><td>-1</td><td>2</td><td>-1</td><td>-1</td></tr></table> | 3 | 8 | 5 | 6 | -1 | -1 | + | x | -1 | -1 | 4 | -1 | -1 | - | -1 | -1 | + | / | -1 | -1 | 2 | -1 | -1 | pré-fixado. |
| 3 | 8 | 5 | 6 | -1 | -1 | + | x | -1 | -1 | 4 | -1 | -1 | - | -1 | -1 | + | / | -1 | -1 | 2 | -1 | -1 | | | |
| C. | <table border="1"><tr><td>6</td><td>x</td><td>4</td><td>+</td><td>-1</td><td>-1</td><td>5</td><td>-</td><td>-1</td><td>-1</td><td>8</td><td>-1</td><td>-1</td><td>/</td><td>-1</td><td>-1</td><td>2</td><td>+</td><td>-1</td><td>-1</td><td>3</td><td>-1</td><td>-1</td></tr></table> | 6 | x | 4 | + | -1 | -1 | 5 | - | -1 | -1 | 8 | -1 | -1 | / | -1 | -1 | 2 | + | -1 | -1 | 3 | -1 | -1 | pós-fixado. |
| 6 | x | 4 | + | -1 | -1 | 5 | - | -1 | -1 | 8 | -1 | -1 | / | -1 | -1 | 2 | + | -1 | -1 | 3 | -1 | -1 | | | |
| D. | <table border="1"><tr><td>6</td><td>x</td><td>+</td><td>/</td><td>-1</td><td>-1</td><td>2</td><td>+</td><td>-1</td><td>-1</td><td>3</td><td>-1</td><td>-1</td><td>5</td><td>-1</td><td>-1</td><td>4</td><td>-</td><td>-1</td><td>-1</td><td>8</td><td>-1</td><td>-1</td></tr></table> | 6 | x | + | / | -1 | -1 | 2 | + | -1 | -1 | 3 | -1 | -1 | 5 | -1 | -1 | 4 | - | -1 | -1 | 8 | -1 | -1 | pós-fixado. |
| 6 | x | + | / | -1 | -1 | 2 | + | -1 | -1 | 3 | -1 | -1 | 5 | -1 | -1 | 4 | - | -1 | -1 | 8 | -1 | -1 | | | |

7. Seja o grafo apresentado na figura seguinte:



Considerando que os pesos das arestas são todos positivos (não nulos), indique todos os valores possíveis para o peso X de forma a que o nó D **SEJA** um nó filho do F na MST com raiz no nó A.

A. {2}.	B. {1, 2}.	C. {2, 3}.	D. {3, 4, 5, ...}.
---------	------------	------------	--------------------

8. Interpretando ‘p’ como a operação abstracta **push** e ‘P’ como a operação **pop**, considere a seguinte sequência de operações sobre uma pilha que armazena caracteres: pppPppPPPpppP. Assuma que cada operação **push** retira um caracter de uma tabela de caracteres, a partir do início da mesma, e que cada operação **pop** é seguida da escrita do caracter removido da pilha. Qual das tabelas abaixo permite produzir a palavra: “razar”?

A. [r a r z a r a r]	B. [z a r z a a r a]	C. [r z a a r r a a]	D. [r a r r a z z r]
----------------------	----------------------	----------------------	----------------------

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.

[5.0]

9. Considere uma sequência de letras do alfabeto (A a Z, incluindo K, W e Y)). A sequência diz-se ter “passo constante” se a distância entre caracteres consecutivos na sequência for sempre a mesma. A distância entre A e B é de 1, entre D e F é 2, P e V é 6, entre Z e C é 3, etc.. As duas sequências abaixo possuem “passo constante”:

ACEG... passo = 2

AZYX... passo = 25

Note que a contagem de distância é circular, onde A é a letra que segue a letra Z.

Duas ou mais sequências dizem-se “entrelaçadas” se os seus elementos adjacentes estiverem separados por exactamente um elemento de cada uma das outras sequências. Por exemplo, as sequências seguintes são “entrelaçadas”:

AQBRCSDT... sequências ABCD... passo = 1

QRST... passo = 1

AZTAYRAXP... sequências AAA... passo = 0

ZYX... passo = 25

TRP... passo = 24

[2.0]

- a) Escreva uma função em linguagem C que receba uma sequência de 12 caracteres e indique quantas subsequências entrelaçadas existem. A assinatura da função é

```
int count_interlaced(char * s);
```

As sequências passadas como argumento possuem sempre comprimento 12 e contêm sempre ou duas ou três subsequências, ou nenhuma. O objectivo da função é apenas determinar quantas existem.

Dadas as sequências acima, esta função deveria retornar 2 para a primeira e 3 para a segunda. A função deverá retornar -1 quando a sequência dada não possui nem 2 nem 3 subsequências entrelaçadas de passo constante.

- [3.0] b) Com base no resultado da função anterior e para cada uma das sequências processadas, escreva outra função que produza os 12 caracteres seguintes, escrevendo-os para `stdio`, juntamente com a sequência original. A assinatura da função deverá ser

```
void complete_sequence(char * s, int n);,
```

em que `n` é o número de sequências entrelaçadas que `s` possui. Caso a sequência dada não corresponda ao padrão esperado, a função deverá escrever apenas a sequência original. Por exemplo:

```
AZTAYRAXPAWN AVLAUJATHASF
ABCDXDCBAXAB
XXYHZRABBLCV DFEPFZGJHTID
```

Nota: Se se pretendesse escrever um programa que lesse cada uma das sequências a partir de um ficheiro e as completasse, a linha de código para completar cada sequência seria `complete_sequence(one_string, count_interlaced(one_string));`.

- [6.0] 10. Considere a matriz de adjacências de um grafo ponderado não direcionado que se apresenta abaixo e assumo que os vértices estão numerados de 1 a 10. Justifique convenientemente os cálculos que efectuar para a resolução das alíneas abaixo.

$$\begin{bmatrix} 0 & 6 & 4 & 8 & \infty & 3 & \infty & \infty & \infty & \infty \\ 6 & 0 & 1 & \infty & 5 & \infty & \infty & \infty & 2 & \infty \\ 4 & 1 & 0 & \infty & \infty & 3 & 9 & \infty & \infty & 3 \\ 8 & \infty & \infty & 0 & 2 & \infty & 5 & \infty & \infty & 10 \\ \infty & 5 & \infty & 2 & 0 & \infty & 2 & \infty & 3 & \infty \\ 3 & \infty & 3 & \infty & \infty & 0 & \infty & \infty & 4 & 7 \\ \infty & \infty & 9 & 5 & 2 & \infty & 0 & 3 & 6 & 5 \\ \infty & \infty & \infty & \infty & \infty & \infty & 3 & 0 & 2 & 5 \\ \infty & 2 & \infty & \infty & 3 & 4 & 6 & 2 & 0 & \infty \\ \infty & \infty & 3 & 10 & \infty & 7 & 5 & 5 & \infty & 0 \end{bmatrix}$$

- [2.5] a) Tomando o vértice 4 como fonte, determine a Árvore de Caminhos mais Curtos por aplicação do algoritmo de Dijkstra.
- [1.0] b) Trace a árvore resultante.
- [2.0] c) Tomando o vértice 6 como fonte, determine a Árvore Mínima de Suporte por aplicação do algoritmo de Prim.
- [0.5] d) Qual o custo da MST que produziu na alínea anterior?

- [3.0] 11. Considere a seguinte tabela palavras da “ThreeLetterWordsLand”, onde todas as palavras são exclusivamente compostas por três caracteres:

[fgt uto pro tyu tat pss jol aar abr cap bba bzt iop iii kal zig gih]

Suponha que pretende transformar aquela tabela num acervo, considerando que uma palavra é mais prioritária que outra se for alfabeticamente anterior. Essencialmente, existem duas formas de construir um acervo a partir de uma tabela: com cópia, transferindo cada elemento da tabela para um acervo inicialmente vazio; ou *in situ*, em que a construção do acervo se faz na própria tabela original.

- [1.25] a) Construa o acervo com cópia.
- [1.25] b) Construa o acervo *in situ*.
- [0.50] c) Os dois acervos assim obtidos são/deveriam ser iguais? Porquê?