

U.C. 21046

Estruturas de Dados e Algoritmos Fundamentais

11 de setembro de 2015

INSTRUÇÕES

- Leia estas instruções na totalidade antes de iniciar a resolução do teste.
- O enunciado do teste é constituído por 4 questões, tem **3** páginas e termina com a palavra **FIM**.
- O teste deve ser resolvido na sua totalidade em folhas de respostas, ficando o aluno com o enunciado.
- O teste é **SEM CONSULTA**. Todos os elementos necessários à resolução são fornecidos no enunciado.
- Utilize esferográfica azul ou preta para responder às questões. Respostas a lápis não serão consideradas.
- Nas respostas, tenha a preocupação de utilizar uma letra legível por outra pessoa.
- A interpretação dos enunciados das questões também faz parte da sua resolução, pelo que, se existir alguma ambiguidade, deve indicar claramente como foi resolvida.
- A correção do teste terá em conta critérios de proficiência e comprehensibilidade do código ou pseudocódigo. Deve assinalar todas as opções tomadas. No código dos seus programas, todas as constantes, variáveis, métodos ou funções devem ser devidamente explicadas através de comentário.
- As respostas, que embora, sintáctica e semanticamente corretas, se apresentem pouco estruturadas serão severamente penalizadas, ou não consideradas. As respostas sem justificação serão fortemente penalizadas.
- Se o seu exemplar não estiver completo ou nele se verificar qualquer outra deficiência, por favor dirija-se ao professor vigilante.
- O não cumprimento das instruções implica a anulação das respetivas questões.
- O tempo de realização do teste é de 120 minutos, mais 30 minutos de tolerância.

1^a Questão (5 Valores)

- a) Construa uma função recursiva que calcula a soma dos primeiros **K** múltiplos de 3. O cabeçalho da função é o seguinte: (2 val)

```
int somaKMultiplos3(int k)
```

- b) Construa uma função iterativa que dado um vector com **N** elementos indica o comprimento da maior subsequência de números pares, em que todos os valores dessa sequência são divisores de **K**. Indique a complexidade do seu algoritmo. O cabeçalho da função é o seguinte: (2 val)

```
int maiorSubSeqParesDivK(int k, int n, int vec[])
```

Ex: Suponha X=[1,4,12,6,24,2,8]; K=3; N=7

somaKMultiplos3(5) = 3+6+9+12+15 = 45

maiorSubSeqParesDivK(K,N,X)=3

- c) Indique a complexidade do seguinte algoritmo justificando. (1 val)

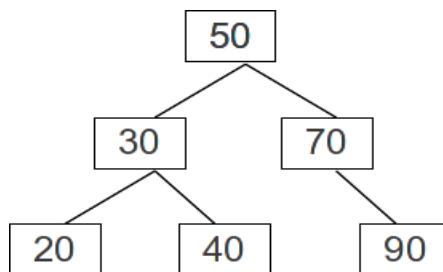
```
int soma=0;  
for(int i=0; i<n; i++)  
    for(int j=i; j<=n;j++)  
        soma+=vec[i][j]
```

2^a Questão (5 Valores)

- a) Desenhe uma árvore binária de altura 3 cuja travessia em pós-ordem (post-order) é a seguinte: (2 val)

1-2-3-4-5-6-7-8-9-10

- b) Considere a seguinte árvore de promoção (Splay Tree). Indique a forma da árvore após o acesso aos seguintes elementos: 50, 30, 40, 70. Deve desenhar a árvore após cada um dos acessos, ou seja faz 4 representações da árvore. (3 val)



3^a Questão (5 Valores)

Considere uma árvore B⁺ de ordem 3 e capacidade 2.

- a)** O que significam os termos **ordem** e **capacidade** numa árvore B⁺ (1 val)
- b)** Indique quais as situações em que o uso de árvores B é bastante recomendado. (1 val)
- c)** Desenhe a árvore após as seguintes operações: Inserção 22, Inserção 16, Inserção 41, Inserção 88, Inserção 8, Inserção 41, Remoção 8. Para cada operação represente a forma da árvore. (3 val)

4^a Questão (5 Valores)

Relativamente aos tipos de dados **Pilha** e **Fila** de inteiros definidos respetivamente pelas seguintes operações. Onde **in**, **inout**, e **out** indicam se o parâmetro é de entrada, entrada-saída e saída respetivamente.

CriaPilha(**out** P:Pilha);
Evazia(**in** P: Pilha) : boolean;
Topo(**in** P: Pilha, **out** N: int);
Push(**inout** P: Pilha, **in** N: int);
Pop(**inout** P: Pilha);

CriaFila(**out** F: Fila);
Evazia(**in** F: Fila): boolean
Insere(**inout** F: Fila, **in** N: int);
Remove(**inout** F: Fila, **out** N: int);

- a)** Implemente em pseudo-código o tipo de dados Fila com 2 pilhas, da forma mais eficiente possível. Por outras palavras, codifique as 4 primitivas sobre filas, assumindo que dispõe apenas de 2 pilhas (P1 e P2) e que só pode recorrer às 5 operações sobre pilhas acima mencionadas. Indique a complexidade temporal dos seus algoritmos, no pior caso. (5 val)

FIM