

## Algoritmos e Complexidade

Exame Normal — 2. Parte — 15 de Junho de 2009

1 – Seja dado um “array” **ordenado** de  $n$  ( $n \geq 2$ ) números reais positivos.

a) Desenvolva uma função eficiente que permita determinar quantos pares de elementos  $(a[i], a[j]), i < j$  e  $j \geq 1$ , existem no “array”, tais que

$$\sum_{k=0}^{i-1} a[k] = a[j].$$

b) Efectue a análise da complexidade do algoritmo da função anterior para o **Melhor Caso**, em termos das adições e das comparações de números reais efectuadas, bem como do número de acessos a elementos do “array”. Identifique instâncias do “array” que conduzam a esse caso.

c) Efectue a análise da complexidade do algoritmo da função anterior para o **Pior Caso**, em termos das adições e das comparações de números reais efectuadas, bem como do número de acessos a elementos do “array”. Identifique instâncias do “array” que conduzam a esse caso.

2 – Seja dado um “array” com  $n$  ( $n \geq 1$ ) elementos, **não ordenado**, sendo cada elemento um número inteiro.

a) Desenvolva uma função **recursiva** que permita determinar o número de vezes que um dado número inteiro ocorre num “array”.

b) Qual é a ordem de complexidade do algoritmo da função anterior? Justifique.

3 – Considere o tipo abstracto de dados *Árvore Binária de Inteiros*, em cujos nós é possível armazenar um número inteiro.

Considere também que os números inteiros **não** se encontram registados em nenhuma **ordem particular**.

Elabore funções que permitam:

a) Obter um *ponteiro* para o nó que contém o *maior elemento* registado numa árvore.

b) Dados uma árvore e dois números inteiros  $a$  e  $b$ , verificar se  $b$  é *descendente* de  $a$  — i.e., se  $b$  pertence a uma das sub-árvores do nó que contém  $a$ .

**Atenção:** deverá ser contemplada a possibilidade de  $a$  e/ou  $b$  não pertencerem à árvore dada.

c) Para cada um dos nós de uma dada árvore, listar — de modo **iterativo** — o número de arcos que definem o correspondente caminho a partir da raiz, ou seja, a sua distância para a raiz.

Os nós deverão ser listados por ordem crescente das distâncias à raiz da árvore.

Por exemplo, para uma árvore com seis elementos ter-se-á:

No	7	-	Distancia	0
No	4	-	Distancia	1
No	5	-	Distancia	1
No	9	-	Distancia	2
No	10	-	Distancia	2
No	11	-	Distancia	3

**Atenção:** assuma que estão definidos os tipos abstractos (genéricos) **Pilha** e **Fila**; não é necessário implementá-los.