



Análise e Desenvolvimento de Algoritmos

Exame Final - 1ª chamada

Data: 5 de Janeiro de 2007

Duração: 2h 30m

Em cada grupo, leia cuidadosamente o enunciado de todas as alíneas, antes de começar a responder.

1. Seja $\mathbb{N}_0$ o conjunto dos números inteiros não-negativos. Para permitir a manipulação de números de grandes dimensões, a representação mais simples consiste num **vector** cujos elementos são os algarismos decimais.

(a) Construa um algoritmo para cumprir a especificação:

Entrada: $\{ n \in \mathbb{N}_0 \}$

Saída: $\{ k \in \mathbb{N}_0 \wedge d[0..k], \forall i \in [0..k] \Rightarrow d_i \in [0..9] :$

$$n = \sum_{i=0}^k d_i 10^i \}$$

(b) Com base na Axiomática de Hoare, verifique formalmente a correcção parcial desse algoritmo.

(c) Pretendemos averiguar se $d[0..k]$ é, ou não, a representação de uma **capicua**. Estabeleça uma especificação formal para o problema.

(d) Elabore um algoritmo para esse efeito e demonstre que cumpre a especificação estabelecida.

2. Sejam $x[1..m]$ e $y[1..n]$ vectores de elementos inteiros. Em cada vector, os números inteiros são todos distintos e estão dispostos por ordem crescente.

(a) Construa um procedimento para realizar a **Fusão Ordenada** desses dois vectores, de modo a que o vector obtido não tenha elementos repetidos.

(b) Faça uma análise do número de comparações efectuadas, envolvendo elementos dos vectores.

(c) Construa um procedimento recorrente para o **Ordenamento por Fusão** de um dado vector de elementos inteiros e distintos.

(d) Calcule o número de comparações efectuadas, envolvendo elementos do vector.

3. Considere o Tipo Abstracto de Dados *Natural*, destinado a manipular números naturais de grandes dimensões. A representação ligada mais simples consiste numa Lista Linear, com um dígito por nó, onde o primeiro elemento contém o algarismo das unidades.

(a) Elabore uma função para construir a representação de um dado $n \in \mathbb{N}_0$:

```
function representacao (n:integer) : natural;
```

(b) Elabore uma função para inverter a ordem dos dígitos de um dado *Natural*:

```
function reverso (n:natural) : natural;
```

(c) Elabore um procedimento iterativo para **destruir** um *Natural*, eliminando correctamente todo o espaço de memória utilizado.

(d) Escreva uma versão recorrente do procedimento anterior.

(e) Escreva uma função para verificar se um dado *Natural* é uma **capicua**:

```
function capicua (n:natural) : boolean;
```