

I (4 v)

1. (1) Sejam  $f, g : \mathbf{N}_0 \longrightarrow \mathbf{R}^+$ , duas funções de custo. Prove que se  $f = \Theta(g)$  então  $f = O(g)$ .
2. (2) Considere uma aplicação que lê uma sequência de inteiros do *standard input* e escreve no *standard output* os  $k$  menores inteiros, onde  $k$  é um parâmetro da aplicação. Descreva um algoritmo para esta aplicação e analise a sua complexidade quanto ao tempo e ao espaço necessário.
3. (1) Esquematize a inserção da seguinte sequência de elementos numa *B-Tree* com  $M = 2T - 1 = 5$  (número máximo de chaves por nó/página):  
 $\{10, 20, 30, 40, 50, 4, 5, 6, 7, 31, 32, 33, 34\}$ .

## II (16 v)

**Nota:** a resolução das questões deste grupo pode utilizar métodos ou classes auxiliares. Contudo, o seu código tem de ser completamente apresentado.

1. (3) Considere o seguinte algoritmo para a obtenção do menor elemento dum *array* de inteiros:

- Se o *array* tiver apenas um elemento, retornar esse elemento.
- Dividir o *array* em dois *sub-arrays*, encontrar o menor elemento de cada *sub-array* e retornar o menor destes dois elementos.

1.1. Implemente um método estático com este algoritmo.

1.2. Analise a complexidade assintótica deste algoritmo.

2. (3) Realize o método estático

```
public static int unique(int[] v, int len);
```

que retira os elementos repetidos do *array* ordenado *v*, com dimensão *len*. No final do método, os elementos presentes no *array* devem estar contíguos. O método retorna o número de elementos presentes no *array*.

3. (3) Realize o método estático

```
public static int minimum(int[] v, int len);
```

que retorna o menor elemento do *max-heap* de dimensão *len*, representado por *v*. O algoritmo deve tirar partido das propriedades de *max-heap* na procura do menor elemento.

4. (3,5) Considere a classe `HashTable<E>` com a implementação de tabelas de dispersão com encadeamento externo e dimensão *N*, onde *N* é uma constante *igual para todas as instâncias da classe*. Realize o método de instância

```
public void move(HashTable<E> otherTable)
```

que adiciona à instância todos os elementos presentes em *otherTable*. A tabela *otherTable* deve ficar vazia.

5. (3,5) Realize o método estático

```
public static int rangeCount(Node root, Integer l, Integer r)
```

que retorna o número de elementos da árvore binária de pesquisa com raiz *root* pertencentes ao intervalo  $[l, r]$ . Assuma que cada objecto do tipo *Node* tem 3 campos: um *value* do tipo *Integer* e duas referências, *left* e *right*, para os descendentes respectivos.