

PUC-Rio  
Departamento de Informática  
Profs. Marcus Vinicius S. Poggi de Aragão  
Período: 2005.1  
Horário: 3as-feiras e 5as-feiras de 15-17  
31 de março de 2005  
Data da Entrega: 31 de maio de 2005

## ESTRUTURAS DISCRETAS (INF 1631)

### 1º Trabalho de Implementação

#### Descrição

Este trabalho prático consiste em desenvolver códigos de algoritmos que ordenam um conjunto de  $n$  inteiros. Os algoritmos a serem codificados são obtidos a partir das provas por indução matemática (simples) do teorema apresentado abaixo.

**Teorema 1** *Sabe-se ordenar um conjunto de  $n$  inteiros  $I^n = \{i_1, i_2, \dots, i_n\}$  onde estes inteiros podem ser representados utilizando-se no máximo  $k$  dígitos.*

Responda às questões que se seguem.

1. Assuma que os inteiros em  $I^n$  são representados na base 10. Apresente a prova do teorema acima por indução matemática utilizando como parâmetro de indução o número de dígitos máximo na representação dos inteiros.
2. Escreva o pseudo-código do algoritmo resultante da prova do item anterior.
3. Repita a prova do item (1) assumindo agora que a base é arbitrária, digamos  $b$ .
4. Modifique o pseudo-código acima para que o algoritmo resultante funcione agora para a representação em bases  $b$  arbitrárias.

Experimentação:

1. Programe o algoritmo obtido para bases arbitrárias.
2. Execute o algoritmo para ordenar os conjuntos de inteiros nos arquivos de teste (disponíveis na página *web* do curso). A especificação dos arquivos de entrada e as saídas desejadas encontram-se mais abaixo.

3. Determine o tempo de CPU gasto para ordenar cada um dos conjuntos (utilize uma função oferecida no ambiente de desenvolvimento).

**Como haverá casos onde o tempo de CPU será muito pequeno, para uma maior precisão, execute o algoritmos tantas vezes quantas sejam necessárias para que o tempo de CPU ultrapasse 5 segundos. Registre o tempo em que o algoritmo terminou sua primeira execução após os 5 segundos de CPU e divida este tempo pelo número de execuções. O valor obtido será o tempo de CPU para ordenar o caso em teste.**

**NÃO INCLUA O TEMPO PARA A LEITURA DO ARQUIVO NO TEMPO DE CPU, CONSIDERE SOMENTE O TEMPO PARA A ORDENAÇÃO !!**

4. Prepare uma tabela com os tempos de CPU obtidos para a ordenação de cada conjunto teste para as seguintes bases: 2, 10, 64, 512, 2048 e 10000.

### Conclusões

Escreva suas conclusões analisando as provas do teorema apresentadas, os algoritmos codificados e os resultados dos experimentos. Você considera que o algoritmo obtido é eficiente ?

### ENTREGA DO TRABALHO

O trabalho pode ser feito em grupos de até **2 (dois)** alunos. O trabalho entregue deve conter:

- Um documento contendo as respostas para as questões acima. Apresente comentários e análises sobre a implementação e os testes realizados (**PAPEL**).
- A impressão do código fonte (papel).
- Um e-mail para **marcus.poggi@gmail.com** contendo o código fonte e o executável correspondente (é obrigatório o uso do ASSUNTO (ou SUBJECT) ED051T1).

### Descrição do Arquivo de Entrada

A primeira linha contém um inteiro  $T$  contendo o número de conjuntos a serem ordenados. Cada um dos  $T$  conjuntos inicia com uma identificação  $I$  do conjunto na primeira linha, na linha seguinte está o número  $N$  de elementos no conjunto, enquanto que as  $N$  linhas seguintes contém os inteiros do conjunto a ser ordenado.

### Descrição do Arquivo de Saída

O arquivo de saída contém as informações para a tabela pedida acima. Ou seja, ele possui  $T$  linhas, onde cada linha apresenta a identificação  $I$  do conjunto, o número  $N$  de inteiros no conjuntos e os tempos de CPU para as bases  $b= 2, 10, 64, 512, 2048$  e  $10000$ .