



Escola de Engenharia da Universidade do Minho
Mestrado Integrado em Eng. Electrónica Industrial e Computadores

2014/2015

MIEEIC

(1º Ano)

2º Sem

Complementos de Programação de Computadores

Luís Paulo Reis

Aula Prática 5: Algoritmos de Pesquisa e Ordenação

Objectivos:

Esta Folha de Exercícios destina-se a:

- Compreender os diversos algoritmos de pesquisa e a noção de complexidade de algoritmos.

Os exercícios aqui propostos deverão ser realizados no mais simples ambiente de desenvolvimento possível para a linguagem C: editor de texto de programação ou editor DevC++ e ferramentas da GCC (GNU Compiler Collection) e afins.

Exercício 5

Supondo o seguinte programa para calcular o tempo decorrido e o número de operações dominantes executadas e as funções simples de criação e escrita de vetores apresentadas:

- 5.1) Construa um template de uma função que gere um vetor aleatório com n elementos.
- 5.2) Construa um template de uma função que imprima no ecrã todos os elementos de um dado vetor.
- 5.3) Escreva um template de uma função para realizar a pesquisa de um valor num vetor utilizando o método de pesquisa sequencial.
- 5.4) Escreva um template de uma função para realizar a pesquisa de um valores num vetor utilizando o método de pesquisa binária.
- 5.5) Adicione aos métodos anteriores capacidade para determinar o tempo decorrido e o número de operações dominantes realizadas na execução das funções de pesquisa.
- 5.6) Escreva um template de uma função para realizar a ordenação do vetor usando ordenação por seleção.
- 5.7) Escreva um template de uma função para realizar a ordenação do vetor usando ordenação por inserção.
- 5.8) Escreva um template de uma função para realizar a ordenação do vetor usando ordenação por partição (quick sort).
- 5.9) Introduza um ciclo que lhe permita calcular os tempos e operações executadas para fazer ambas as pesquisas no vetor ordenado para dimensões entre 1 e 10000, gravando os resultados obtidos num ficheiro.
- 5.10) Construa gráficos (por exemplo utilizando a ferramenta Excel) para os tempos e números de operações, comparativos da eficiência de todos os métodos de pesquisa e ordenação.

```

// Programa Exemplo de Contagem de Operações e de Tempo

#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <ctime>

using namespace std;

long int op;          //Contador de Operações
#define N 10000       //Dimensão dos Dados

double diffclock(clock_t clock1, clock_t clock2)
{
    return (double)(clock1-clock2)/CLOCKS_PER_SEC;
}

void cria_vetor(int v[], int n)
{
    for(long int i=0; i<n; i++)
        v[i] = rand() % 20000;
}

void escreve_vetor(int v[], int n)
{
    for(long int i=0; i<n; i++)
        cout << v[i] << " ";
    cout << endl;
}

int main()
{
    string name;
    long int i;
    clock_t begin, end;

    int v[N];
    cria_vetor(v, N);
    escreve_vetor(v, N);

    /*cout << "Qual e' o teu Nome? ";
    begin=clock();
    getline(cin, name);
    end=clock();
    cout << "Tempo: " << diffclock(end, begin) << " seg"<< endl; */
    /* begin=clock();

    op = 0;
    for(i=0; i<50000000; i++) {
        float x= i/3.0; //gastar tempo
        op++;
    }
    end=clock();
    cout << "Tempo: " << diffclock(end, begin) << " seg"<< endl;
    cout << "Operacoes: " << op << endl; */

    system("PAUSE");
    return 0;
}

```