UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Bacharelado em Ciência da Computação
Trabalho Prático de Estrutura de Dados - Ordenação
Prof. Roberto Cabral

1 Descrição

Este trabalho consiste em analisar o desempenho de diferentes algoritmos de ordenação em diferentes cenários, descritos a seguir. Esta análise consiste em comparar os algoritmos considerando três métricas de desempenho:

1. Número de comparações de chaves;
2. Tempo total gasto para ordenação (tempo de processamento e não o tempo de relógio).

As entradas deverão ser conjuntos de elementos com chaves aleatoriamente geradas. Para obter o tempo de processamento na linguagem C, você pode utilizar o comando getrusage, lembrando de somar os tempos gastos em modo de usuário (utime ou user time) e em modo de sistema (stime ou system time) (vide exemplo no final do enunciado).

2 Cenário 1: Impacto de diferentes estruturas de dados

Neste cenário, você deverá avaliar o desempenho do método de ordenação Quicksorte (recursivo) considerando as seguintes condições:

1. O método deve ser executado em estruturas de dados linear **estática** e **dinâmica** com 5 mil elementos;

2. Primeiramente, o método deve ordenar inteiros escolhidos de forma aleatória nas duas estruturas. A mesma posição de cada estrutura recebe o mesmo inteiro. Ao final, as comparações propostas na descrição do trabalho devem ser mostradas e os dados deverão ser inseridos em um txt.

3. Após a primeira etapa, o método deve ordenar caracteres escolhidos de forma aleatória nas duas estruturas. A mesma posição de cada estrutura recebe o mesmo caractere. Ao final, as comparações propostas na descrição do trabalho devem ser mostradas e os dados deverão ser inseridos em um txt.

3 Cenário 2: Quicksort X Mergesort X Heapsort X SelectionSort X InsertionSort X BubbleSort

Neste cenário, você comparar diferentes estratégias de ordenação para ordenar um conjunto de N inteiros positivos, aleatoriamente gerados, armazenados em um vetor. Realize experimentos considerando vetores aleatoriamente gerados com tamanho N = 1000, 5000, 10000, 50000, 100000, 500000, 1000000, no mínimo. Para cada valor de N , realize experimentos com 5 sementes diferentes. Para a comparação dos algoritmos de ordenação, avalie os valores médios do tempo de execução, número de comparações de chaves e do número de cópias de valores. Apresente gráficos e/ou tabelas com os resultados obtidos. Discuta os resultados e conclusões obtidas, considerando as diferentes métricas. Qual algoritmo tem melhor desempenho. Por quê?

4 Medindo o tempo de execução de uma função em C:

O comando getrusage() é parte da biblioteca padrão de C da maioria dos sistemas Unix. Ele retorna os recursos correntemente utilizados pelo processo, em particular os tempos de processamento (tempo de CPU) em modo de usuário e em modo sistema,fornecendo valores com granularidades de segundos e microssegundos. Um exemplo que calcula o tempo total gasto na execução de uma tarefa é mostrado abaixo:

```
#include <stdio.h>
#include <sys/resource.h>

void main () {
    struct rusage resources;
    int rc;
    double utime, stime, total_time;

    /* coloque o procedimento a ser medido aqui */

    if((rc = getrusage(RUSAGE_SELF, &resources)) != 0)
        perror("getrusage failed");

    utime = (double) resources.ru_utime.tv_sec
        + 1.e-6 * (double) resources.ru_utime.tv_usec;
    stime = (double) resources.ru_stime.tv_sec
        + 1.e-6 * (double) resources.ru_stime.tv_usec;
    total_time = utime+stime;

    printf("User time %.3f, System time %.3f, Total Time
       %.3f\n", utime, stime, total_time);
}
```

tempo.c

OBS: modificação do trabalho disponível em:
http://www2.dcc.ufmg.br/disciplinas/aeds2_turmaA1/tp3-2009-2.pdf

Bom trabalho!