



Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação

Disciplina Programação (CK0226)
Tarefa de Laboratório nº 05
Experimento em Algoritmos de Classificação

Prof. Miguel Franklin

OBJETIVOS:

- ☐ Familiarização com a sintaxe e as estruturas da linguagem C
- ☐ Prática de manipulação de vetores e algoritmos de classificação

ENUNCIADO:

Faça um programa em linguagem C que implemente quatro funções de classificação de um vetor de N posições de inteiros: Bolha (http://pt.wikipedia.org/wiki/Bubble_sort), Insert Sort (http://pt.wikipedia.org/wiki/Insertion_sort), Selection Sort (http://pt.wikipedia.org/wiki/Selection_sort) e Comb Sort (https://pt.wikipedia.org/wiki/Comb_sort). O programa deve realizar o seguinte procedimento K vezes, para cada algoritmo de classificação: (1) povoar um vetor de N posições com elementos inteiros aleatórios no intervalo entre $[0; 100 * N]$. Não é necessário checar se existe elementos duplicados. (2) Classificar o vetor. O tempo de execução dessas K classificações deve ser medido para cada algoritmo. No final, o programa deve mostrar o tempo de execução das K classificações, para cada algoritmo de classificação, mostrando, assim, qual foi o mais eficiente.

Você deve ajustar, experimentalmente, os valores de N e de K para que o algoritmo mais eficiente conclua a tarefa em cerca de 4 (quatro) segundos. Esse ajuste experimental é necessário pois o desempenho dos algoritmos será, obviamente, bastante dependente da potência de cálculo do computador que estiver sendo usado.

Dica: É possível recuperar as implementações dos algoritmos de Bolha e *Selection Sort* que foram feitas na Tarefa de Laboratório passada.

A entrega (upload) deverá ser realizada através do Google Classroom.

ANEXOS

Exemplo para geração de números randômicos em C:

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 int main(int argc, const char * argv[]) {
6     int i, num_rand;
7
8     srand((int)time(NULL)); // Define a semente de geração de número randômico para o clock do computador
9                             // O srand() só precisa ser chamado uma vez, no início do programa.
10
11     for(i = 0; i < 100; i++)
12         printf("Numero pseudorandomico: %ld\n", (random() % 100001)); // Imprime número randômico no intervalo [0; 100000]
13     return 0;
14 }
```

Exemplo para a contabilização do tempo de execução:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/time.h> // <<<--- ATENÇÃO AQUI
4
5 int main(void) {
6     struct timeval inicio, fim;
7     double tempo_em_segundos = 1.0;
8
9     // Variáveis do cálculo do PI
10    int r[2800 + 1];
11    int i, k;
12    int b, d;
13    int c = 0;
14
15    gettimeofday(&inicio, NULL);
16
17    /* Colocar aqui o processamento que se quer fazer */
18    for (i = 0; i < 2800; i++) {
19        r[i] = 2000;
20    }
21    r[i] = 0;
22
23    for (k = 2800; k > 0; k -= 14) {
24        d = 0;
25
26        i = k;
27        for (;;) {
28            d += r[i] * 10000;
29            b = 2 * i - 1;
30
31            r[i] = d % b;
32            d /= b;
33            i--;
34            if (i == 0) break;
35            d *= i;
36        }
37        printf("%.4d", c + d / 10000);
38        c = d % 10000;
39    }
40
41    gettimeofday(&fim, NULL);
42
43    // Calcula o tempo que levou para fazer o processamento
44    tempo_em_segundos = (double) (fim.tv_usec - inicio.tv_usec) / 1000000 + (double) (fim.tv_sec - inicio.tv_sec);
45
46    printf ("Tempo total de processamento = %f segundos\n", tempo_em_segundos);
47
48    return EXIT_SUCCESS;
49 }
```