

Lab_05

//-----

Prazo de entrega: 31/05/2025 às 23:59h

//-----

Exercício 2, livro: Introdução a Estruturas de Dados com Técnicas de Programação em C - Celes, Cerqueira, Rangel, 2ª edição.

Implemente um código para fazer uma análise do desempenho de funções de procura, comparando procura binária em vetor e procura em tabela de dispersão.

O experimento pode ser feito usando valores inteiros: buscar um valor inteiro em um vetor ordenado versus buscar um valor inteiro em uma tabela de dispersão. Considerando que n representa o número de valores inteiros, crie um vetor desta dimensão e uma tabela de dispersão (pode-se já predefinir a tabela com o dobro do tamanho para evitar redimensionamento).

Gere aleatoriamente n valores inteiros usando a função `rand`; insira o valor gerado no vetor e na tabela de dispersão; ao final, ordene o vetor. Então, repita a geração dos n valores. Para garantir que a mesma sequência de valores seja gerada, use um mesmo valor de semente (por exemplo, chamando `srand (0)` antes de gerar o primeiro número de cada sequência).

Para cada valor, busque o valor no vetor usando procura binária. Para medir o tempo gasto para fazer a procura dos n valores, use a função `clock` da biblioteca `time .h`, que reporta o tempo de CPU gasto desde a invocação do processo (isto é, desde o início da execução do programa), medidos em clocks por segundo (`CLOCKS_PER_SEC`, uma constante também definida em `time .h`).

Interessa-nos a diferença de tempo de CPU entre o início e o fim do procedimento que queremos medir. Em seguida, repita o mesmo procedimento, gerando e procurando os n valores agora na tabela de dispersão, medindo o tempo de execução. Imprima e compare os tempos de busca de cada estratégia. Use diferentes valores de n e verifique a eficiência da tabela de dispersão. O trecho de código a seguir ilustra a tomada de tempo de execução de um procedimento.

```
clock_t t0 = clock ();
```

```
... /* procedimento que se deseja medir tempo de  
execução */  
clock_t t1 = clock();  
double t = (double)(t1 - t0) / CLOCKS_PER_SEC; /* tempo (seg) */
```

Este exercício deverá ser construído com modularização e deverá compilar e funcionar tanto no Linux quanto no Windows, sem modificações. Deverá funcionar nos computadores da escola.

O grupo deverá escrever um relatório de, no máximo, 3 páginas, contendo: introdução, objetivo, resultados, discussão dos resultados e conclusão.