

UNIVERSIDADE DO ESTADO DE MINAS GERAIS
UNIDADE DIVINÓPOLIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO
ALGORITMOS E ESTRUTURA DE DADOS 2
PROFESSOR: EDWALDO SOARES RODRIGUES
DOCUMENTAÇÃO
TRABALHO PRÁTICO 2 - MÉTODOS DE ORDENAÇÃO

Wendell Resende dos Santos

1- Objetivo do Programa

O programa consiste em demonstrar quais Métodos de Ordenação, dentre os usados é mais eficiente. Foram usados os algoritmos de ordenação Insert Sort, Selection Sort, Merge Sort e Quick Sort.

2- Resumo do Programa

O programa conta com uma struct Aluno, a qual possui dados relevantes sobre alunos e tem como sua chave principal a matrícula. Foi então armazenado em um vetor de tamanho n, todos os alunos, e posteriormente ordenados pelos 4 métodos de ordenação, após sair de algum o vetor se desfaz, se é criado outro, e este é ordenado por outro método de ordenação. Isso se repete para valores de n, iguais a 1000, 10000, 100000 e 200000.

Após essa primeira ordenação, novamente são ordenados, porem agora já estavam previamente ordenados. E após essa segunda parte, ele é ordenado novamente, porem dessa vez ele estava inversamente ordenado. O programa dispõe de um temporizador para a verificação de quanto tempo leva cada ordenação por cada vetor em específico.

3- Objetivo do Programa

O programa tem por objetivo principal analisar o tempo de execução de cada método, medindo seu tempo de execução para diferentes tamanhos de vetores, podendo assim chegar a uma conclusão prática, sobre qual método utilizar em cada caso do cotidiano do programador.

4- Dados do programa em específico

O programa se inicia e então logo é gerado um arquivo txt, o qual será nosso controle de código, lá ficarão todos os tempos de execução de cada algoritmo utilizado. É ordenado então um vetor de alunos em cada um dos 4 métodos, para os 4 tamanhos de vetores. Dessa forma é possível de analisar a eficácia dos métodos em diferentes tamanhos e com vetores desordenados aleatoriamente. Após essa primeira parte o programa repete o processo, porem agora o vetor Aluno, está ordenado, e é possível obter resultados sobre a ordenação quando os vetores já estão organizados. Após esse processo, ocorre novamente a ordenação, porem agora o vetor está inversamente ordenado. Ao longo do programa o usuário pode escolher ver ou não o vetor, antes ou depois de ordenar.

5- Sobre os Métodos utilizados

Abaixo estão os 4 métodos utilizados no programa.

- **Seletion Sort**

Esse primeiro programa seleciona um elemento no vetor e se caso encontrado outro de maior valor, eles trocam de posição.

- **Insertion Sort**

Um pouco mais complexo que o anterior, este percorre o vetor pegando o primeiro elemento após a parte do vetor ordenado, enquanto não encontra um elemento de maior valor, ele ocupa o lugar deste, fazendo com o vetor se ordene.

- Merge Sort

Esse método é recursivo, isso significa que ele mesmo chama a si próprio passando como argumento uma parte do vetor principal, essa parte sempre será a metade do vetor que ele está recebendo, seja do main, ou dele próprio através da recursão. O método consiste em dividir um vetor de tamanho n , em vetores de tamanho 1, e voltar da recursão ordenado de par em par os vetores, ate novamente voltar ao main, com o vetor ordenado.

- Quick Sort

Esse se assemelha em alguns pontos com o Merge Sort, utilizando também a recursão. Porem neste, é encontrado um pivô no centro do vetor que a função recebe, e dentro deste, os elementos a esquerda do pivô, que são maiores que ele, trocam de lugar com o elemento menor que o pivô do outro lado do vetor. O vetor é percorrido, de um lado tendo acréscimo no índice, e de outro um decremento, quando os índices se encontram ocorre a recursão. O vetor é ordenado, quando se volta da recursão e todos os subvetores do vetor original estão devidamente ordenados.

6- Conclusão

Foi identificado que sempre um será mais rápido ou mais lento que outro, tendo que ocorrer uma análise prévia do problema a ser resolvido, para se poder optar pelo melhor procedimento. A eficácia de todos é evidente, mostrando que todos funcionam como esperado. Esse trabalho foi de grande valia para a avaliação prática dos métodos vistos em sala de aula, podendo assim tirar conclusões, na prática, de como e quando utilizar cada um.

7- Dificuldades ao realizar a verificação

Ocorreu diversos problemas ao executar o código, todos contornados. Porem o programa funciona perfeitamente com vetores pequenos, mas não consegue ordenar vetores grandes. O arquivo é sobrescrito corretamente passando os valores necessários e é possível analisar alguns casos de tamanhos de vetor, porem não da forma e tamanho que o trabalho pedia.

8- Bibliografia

<https://stackoverflow.com/search?q=Sorting+method+in+C>

<https://www.w3schools.com/>