

Busca e Ordenação

Trabalho 2 - valor 5

Professor: Tadeu Zubaran

1 Introdução

O trabalho deve ser feito ou em C++(recomendado) ou em C. Você deve criar um Insertion Sort e um Selection Sort. Você deve usar uma lista por contiguidade física. Essas listas armazenam valores inteiros. Utilize apenas a memória auxiliar necessária.

Você pode escolher o nível de controle de erros e inconsistências que quiser. O uso de orientação a objetos também é facultativo.

2 Retorno

Ao final da execução o seu algoritmo deve mostrar a lista ordenada.

Você deve ter a opção de:

1. Mostrar o estado da lista a cada iteração do algoritmo, i.e. ao final de cada iteração do algoritmo, ou seja após uma inserção ou seleção.
2. Mostrar o número de acessos à estrutura de dados (você pode considerar tanto escrita como leitura como acesso) ao final da execução do algoritmo.

3 Testes

3.1 Ler lista

Execute seu algoritmo para a lista:

1 70 1 12 8 99 72 5 15 20 91 14 61 66 41 81 88 16 21 34 90 Utilize a opção 1, que mostra

o estado da lista a cada iteração do algoritmo.

3.2 Lista aleatória

Execute o seu algoritmo (serão 100 vezes no total), cada vez um em lista diferente aleatória com um tamanho n diferente, onde $n \in [1, 100]$.

- O número de elementos da lista é n , e os
- intervalos de valores contidos na lista é $[1, 1000]$

Execute seu algoritmo com a opção 2 que mostra o número de acessos à memória, e guarde essa informação. Faça um gráfico no qual o

- eixo vertical mostra número de acessos à memória, e o
- eixo horizontal mostra o número de elementos da lista.

Você pode usar o software da sua preferência para fazer esse gráfico.

Mostre e EXPLIQUE esse gráfico em sala de aula. Deixe muito claro na entrega do seu código como reproduzir os testes que geram seu gráfico com o código que você enviou.

4 Entrega

A entrega consiste de duas coisas. Upload do código no google class e apresentação do código em aula. Trabalhos sem upload ou sem apresentação em aula não serão considerados. Lembre que o dia de apresentação é um dia de avaliação.

O upload do código deve ser feito no class. Faça o upload apenas do seu *.cpp (ou *.c) e de seus *.h (ou *.hpp). Caso você faça seu próprio makefile faça o upload dele também. Não compacte, você pode fazer upload de mais de um arquivo em cada tarefa. ENvie qualquer arquivo necessário para reprodução dos gráficos.

5 Avaliação

Não copie o código. Faça seu próprio código! Plágio receberá nota 0.

Critérios de avaliação:

- Clareza e corretude da explicação do código.
- Clareza e corretude do código.
- Explicação das decisões de projeto.

- Código implementado com boas práticas de
programação. 2