
ALGORITMOS DE ORDENAÇÃO: ANÁLISE COMPARATIVA

Considere a implementação em C das seguintes funções de ordenação:

- Inserção Direta;
- Shell Sort;
- Bubble Sort;
- Quick Sort;
- Heap Sort ;
- Merge Sort .

OBS.: O único algoritmo que não tem a implementação no moodle é o HeapSort. Você pode pegar uma implementação disponível na Web ou implementar a partir dos slides disponíveis no moodle.

PREPARAÇÃO DO EXPERIMENTO:

Prepare um ambiente experimental para avaliar os algoritmos de ordenação da seguinte forma:

Ambiente 01 (Caso Médio). Use um gerador de números aleatórios para gerar os arquivos (vetores) de tamanho 500, 5.000, 10.000 e 30.000 elementos. Para cada arquivo utilize dois tamanhos de registros: **(i)** um contendo apenas a chave; e **(ii)** outro de tamanho 11 vezes o tamanho da chave (ou seja, a chave acompanhada de “outros 11 componentes (atributos)” cujo tamanho seja equivalente a 11 chaves).

Ambiente 02 (Melhor Médio). Repita o mesmo experimento do ambiente 01, agora, com os vetores ordenados em ordem **crecente**:

Vetor = {1 .. 500}
Vetor = {1 .. 5000}
Vetor = {1 .. 10000}
Vetor = {1 .. 30000}

Ambiente 03 (Pior Médio). Repita o mesmo experimento do ambiente 01, agora, com os vetores ordenados em ordem **decrecente**:

Vetor = {500 .. 1}
Vetor = {5000 .. 1}
Vetor = {10000 .. 1}
Vetor = {30000 .. 1}

EXECUÇÃO DOS EXPERIMENTOS:

1 – Executar cada um dos algoritmos cinco vezes, calculando, para cada um dos quatro conjuntos de dados, o tempo gasto para ordenação e ao final calcular a média de tempo gasto. Ao final, dê a sua interpretação para os resultados obtidos.

Exemplo de Documentação:

Número Elementos	Execução	Inserção Direta	Shell Sort	Bubble Sort	...
500 registro com uma chave	Execução 01				
	Execução 01				
	Execução 01				
	Execução 01				
	Execução 01				
	Média do tempo gasto				
...	...				

Interpretação dos resultados: ...

2 – Executar cada algoritmo novamente. Agora o algoritmo deve ser executado apenas uma vez para cada tamanho de registro (chave / chave + 11 atributos) em cada um dos ambientes. Você deve contabilizar:

- número de comparações;
- número de movimentos de registros (copia/transferência de registro de uma posição para outra).

Ao final, dê a sua interpretação para os resultados obtidos.

3 – Considerando os resultados do seu experimento, responda:

- a) Qual o melhor algoritmo para arquivos pequenos (ex. 500 elementos)? Por quê?
- b) Qual o melhor algoritmo para arquivos grandes (ex. 30.000 elementos)? Por quê?
- c) Qual o melhor algoritmo para arquivos com registros extensos, ou seja, registros com muitos atributos (ex. registros com chave + 11 atributos)? Por quê?
- d) A medida em que o número de elementos cresce, qual o melhor algoritmo a ser utilizado? Isto é, quanto maior o número de elementos no arquivo o algoritmo melhora seu desempenho.
- e) No caso de se utilizar um único algoritmo para todas as situações dos experimentos, qual algoritmo você escolheria? Por quê?

ORGANIZAÇÃO DO TRABALHO:

Entrega:

Data: 05/05 – quarta-feira

Horário: 13 hrs

O prazo não será prorrogado!

Grupos: o trabalho pode ser desenvolvido em grupos de, no máximo, 3 alunos.

Material Entregue: submeter um arquivo compactado no moodle contendo:

- implementação desenvolvida;
- documento do tipo texto contendo a descrição dos resultados obtidos nos itens 1 e 2 e as respostas para o item 3.