

Trabalho Prático 2

Laboratório de Algoritmos e Estruturas de Dados I
Prof^a. Natália Batista

Nome: _____ Total: 15 pontos

Instruções

O trabalho prático é individual e deverá ser entregue até o dia 10/11/2017, por *email*. O prazo de entrega deste trabalho NÃO será prorrogado.

Este trabalho deverá conter uma documentação conforme a especificação do Anexo I. A parte prática deverá ser implementada em linguagem C ou C++ no ambiente Linux. Lembre-se de incluir as instruções de compilação dos programas e eventuais arquivos auxiliares, como “makefiles”.

Para a entrega dos arquivos (documentação e códigos), siga as instruções contidas no Anexo II.

O enunciado deste trabalho foi adaptado de um dos trabalhos da disciplina de Projeto e Análise de Algoritmos do Prof. David Menotti Gomes (DECOM/UFOP).

Implementação e Análise de Algoritmos de Ordenação por Comparações de Chaves

1. Descrição

Neste trabalho, você deverá implementar algoritmos não-eficientes (BubbleSort, InsertionSort e SelectionSort) e eficientes (QuickSort, HeapSort e MergeSort) para ordenar sequências de elementos.

Você deverá utilizar implementações dos algoritmos citados onde a sequência de dados a ser ordenada pode estar representada por arranjos (vetores) ou apontadores. Lembre-se que as implementações dos algoritmos de ordenação estão disponíveis nos slides de aula da disciplina teórica (https://sites.google.com/site/nataliacefetmg/home/aed1_20172) e deverão ser utilizadas. Todavia, o domínio e conhecimento das implementações podem ser objeto de avaliação em entrevista.

Cada elemento da sequência de dados a ser ordenada deve conter um campo chave e pelo menos um campo adicional. A especificação e desenvolvimento dos tipos abstratos de dados a serem utilizados no trabalho deverão ser baseados nas implementações do livro texto da disciplina (<http://www2.dcc.ufmg.br/livros/algoritmos/implementacoes.php>).

2. Como gerar os vetores/arranjos para ordenação

Considere sequências de dados com a quantidade de elementos variando, i.e., 10, 100, 1000, 10000, 100000, 1000000, etc. Considere também sequências sem valores repetidos. Ainda, considere que todos os elementos das sequências correspondem a valores inteiros e para gerar as sequências iniciais utilize sequências **ordenadas**, **inversamente ordenadas**, **quase ordenadas** e **aleatórias**.

3. O que analisar

A análise deverá ser feita sobre o número de comparações, atribuições e tempo de execução dos algoritmos. Procure organizar inteligentemente os dados coletados em tabelas, e também construa gráficos a partir dos dados. Então, disserte sobre os dados nas tabelas e gráficos. Grande parte da avaliação será dedicada a análise dos resultados, ou seja, sobre o que você dissertar.

Para comparação, relacione somente os algoritmos não-eficientes e eficientes entre si. Por exemplo, não relacione, analise ou compare, o algoritmo BubbleSort com o QuickSort.

4. O que deverá ser entregue

- Código fonte do programa em C ou C++ (bem indentado e comentado).
- Documentação do trabalho.

A documentação deverá conter, além dos itens descritos no Anexo I:

1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
2. Implementação: descrição sobre a implementação do programa. Deverão ser detalhadas as estruturas de dados utilizadas (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Muito importante: os códigos utilizados nas implementações deverão ser inseridos no final da documentação como Anexo.
3. Análise de Complexidade: estudo da complexidade de tempo e espaço das funções implementadas e do programa como um todo (notação O).
4. Listagem de testes executados: os testes executados devem ser apresentados, analisados e discutidos.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
6. Referências: toda bibliografia utilizada para o desenvolvimento do trabalho, incluindo páginas da Internet, se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização.
8. Formato final: mandatoriamente em PDF (

5. Como deverá ser feita a entrega

A entrega deverá ser feita por email na forma de um único arquivo zipado (.ZIP), contendo o código fonte, arquivos diversos e a documentação. As instruções de entrega estão no Anexo II.

Também deverá ser entregue a documentação impressa até o dia 14/11.

6. Comentários Gerais

- O trabalho será realizado durante as aulas práticas dos dias 20/10, 27/10 e 10/11.
- Clareza, identificação e comentários no programa também serão avaliados;
- O trabalho é individual;
- Trabalhos contendo evidências de plágio terão nota zero.
- Trabalhos entregues em atraso não serão aceitos;
- A data de entrega é fixa.