

Algoritmos e Estruturas de dados 2

Trabalho Final

6 de novembro de 2024

1 Objetivos e organização

Esta tarefa possui como objetivo:

- Desenvolver um sistema de gerenciamento de uma biblioteca universitária utilizando estruturas de dados avançadas para otimizar as operações de busca, inserção e organização dos dados.

Os programas implementados serão apresentados ao professor da disciplina que os avaliará.

Trabalhos não apresentados **não** serão avaliados. Os códigos fontes deverão ser encaminhados para mim via a equipe Teams da disciplina.

No dia da apresentação **todos** os membros do grupo devem estar presentes e serão arguidos individualmente.

Todo o trabalho deve ser documentado. Assim, os códigos das funções definidas por você devem conter, entre outras informações, o nome e o propósito de cada algoritmo implementado e o código desse algoritmo. Além disso, ele deve conter instruções específicas sobre como compilar e executar os programas.

O trabalho é em **dupla** e qualquer desonestidade acadêmica será punida com nota **zero** para todos os envolvidos.

1.1 Informações importantes

1.1.1 Sobre o código

- Modularize o seu programa, ou seja, crie várias funções.
- Evite criar funções que ocupem mais que 50 linhas de texto.
- Mantenha a sua função **main** com um mínimo de chamadas de funções e de código.

- O código deve ser desenvolvido em C padrão (C11 ou C17)
- Usar alocação dinâmica de memória
- Implementar tratamento de erros
- Documentar o código seguindo padrão Doxygen
- Criar Makefile para compilação

1.1.2 Detalhes da entrega

apresentação 19 de novembro de 2024 das 14h até as 15h, terça-feira, na sala 1B123.

entrega 19 de novembro de 2024 até as 17h, terça-feira. O horário limite de entrega, em nenhuma situação, poderá ser ultrapassado. Você pode antecipar a apresentação se terminar antes desse dia.

mídia O código fonte deve ser enviado via Teams. Coloque os nomes de todos os membros do grupo.

2 Enunciado

Este trabalho consiste em escrever um programa em C para o gerenciamento de uma biblioteca universitária utilizando estruturas de dados avançadas para otimizar as operações de busca, inserção e organização dos dados.

3 Requisitos do Sistema

Para todas as operações pedidas crie um sistema de menus para facilitar o acesso dentro de seu sistema. O menu pode ser baseado em texto.

3.1 Parte 1: Árvore AVL (5 pontos)

1. Implementar uma árvore AVL para armazenar livros com os seguintes campos:
 - Código ISBN (chave)
 - Título
 - Autor

- Ano de publicação
 - Quantidade disponível
2. Implementar as seguintes operações:
 - Inserção de novo livro
 - Remoção de livro
 - Busca por ISBN
 - Balanceamento automático
 - Exibição da árvore em pré-ordem, in-ordem e pós-ordem
 - Carregar dados de arquivo texto.
 - Salvar dados para arquivo texto.

3.2 Parte 2: Grafo de Relacionamentos (8 pontos)

1. Implementar um grafo utilizando lista de adjacências para representar relações entre livros:
 - Vértices são livros
 - Arestas representam relações (mesmo autor, mesmo gênero, etc.)
 - Peso das arestas indica força da relação (1-5)
2. Implementar os seguintes algoritmos:
 - Busca em profundidade (DFS)
 - Busca em largura (BFS)
 - Dijkstra para encontrar o caminho mais **forte** entre dois livros.
Note que é um problema de encontrar o caminho de custo máximo.
 - Função para sugerir livros relacionados
 - Carregar dados de arquivo texto.
 - Salvar dados para arquivo texto.

3.3 Parte 3: Algoritmos de Ordenação (3 pontos)

1. Implementar três algoritmos de ordenação:
 - QuickSort
 - HeapSort

- MergeSort
2. Criar funções para ordenar os livros por:
 - ISBN (com QuickSort)
 - Título (com HeapSort)
 - Ano de publicação (com MergeSort)

4 Critérios de Avaliação

- Compilação sem erros (10%). Se não compilar a nota é **zero**.
- Corretude das implementações (50%)
- Eficiência das soluções (20%)
- Qualidade do código e documentação (20%)

5 Dicas de Implementação

1. Comece implementando as estruturas básicas (nós da árvore, vértices do grafo)
2. Teste cada função individualmente
3. Use arquivos .h para declarações e .c para implementações
4. Mantenha um log de testes

6 Bônus (5 pontos)

- Implementar interface gráfica para visualização das estruturas
- Implementar balanceamento Rubro-Negro como alternativa à AVL, ou seja, além da AVL.