

Algoritmos e Estruturas de Dados III - Árvores Red-Black

Paulo Ricardo Lisboa de Almeida

Agosto de 2025

1 Descrição

O programa a seguir servirá como base para a prova prática da disciplina. Caso você não traga o programa, ou se o programa não for de sua autoria (por exemplo, por plágio ou uso de IA Generativa), sua nota na prova será automaticamente zero. Além das questões que serão reveladas apenas durante a prova, a qualidade do seu programa terá grande peso na avaliação.

O Objetivo é implementar uma árvore Red-Black utilizando a linguagem de programação C. A implementação deve seguir exatamente a descrita em Cormen et al. (algoritmos discutidos em sala). Detalhes são dados a seguir.

2 Template

Os algoritmos devem ser implementados utilizando como base o template disponibilizado na UFPRVirtual.

O arquivo *redblack.h* contém os protótipos das funções que obrigatoriamente devem ser implementadas, no arquivo *redblack.c*.

Não é permitida a mudança nos protótipos de funções disponibilizados no arquivo *redblack.h*. É permitida a criação de funções auxiliares se necessário. Também não é permitida a criação de arquivos *.h* ou *.c* extras.

O nome do binário gerado deve ser *antesprova2_GRR* (e.g., *antesprova2_1234*). Ajuste isso no *makefile*. Deve ser possível compilar e executar o binário nas máquinas do DInf.

3 Detalhes de implementação

As seguintes regras devem ser obedecidas:

- A chave dos nodos deve ser um inteiro.
- O programa deve compilar sem erros ou avisos através do comando *make*.
- A árvore não deve aceitar chaves duplicadas. Ao tentar inserir uma chave duplicada, a função de inserção deve retornar SENTINELA.
- Uma árvore vazia é uma árvore em que a raiz aponta para SENTINELA.

- Garanta que os custos dos algoritmos implementados é compatível com a teoria sobre árvores Red-Black. Por exemplo, inserções, exclusões e buscas devem ter um custo $O(\log_2 n)$, enquanto a impressão da árvore deve ter um custo $O(n)$. Algoritmos ineficientes serão desconsiderados.
- Ao faltar memória (em um `malloc`, por exemplo), o programa deve ser terminado imediatamente através da chamada `exit(1)`. Exiba a mensagem de erro *Falha ao alocar memória.* em `stderr`.

4 Qualidade do programa

Espere descontos severos por programas que não seguem boas práticas e afins. Exemplos de problemas incluem funções que exibem erros na tela ao invés de retornar códigos de erro, uso de tipos de dados incorretos, códigos obscuros e afins.

Vazamentos de memória e codificações com brechas de segurança (exemplo: *buffers overflows* devido ao uso incorreto de `scanf` e afins) sofrerão os maiores descontos.

5 Formato de Entrada e Saída

A entrada será feita pela entrada padrão (`stdin`). A entrada será uma sequência de linhas, onde cada linha representa uma operação. As entradas podem ser as seguintes:

- inserir um valor na árvore: `i val`
- remover um valor na árvore: `r val`
- buscar um valor na árvore: `b val`
- imprimir a árvore em ordem: `e`
- imprimir a árvore em largura: `l`
- finalizar e sair: `f`

Por exemplo, *i 10* significa a inclusão da chave 10 na Árvore.

A saída será feita pela saída padrão (`stdout`). Saídas são geradas apenas quando o usuário solicitar para imprimir a árvore, ou buscar um item, ou em caso de erro. Veja um exemplo de execução do programa a seguir, onde os itens em *itálico* são entradas, e em **negrito** são saídas geradas pelo programa.

i 50

i 40

i 35

i 35

Falha ao inserir.

i 45

l

[0] (B)40 [QUALQUER]

[1] (B)35 [40e] (B)50 [40d]

[2] (R)45 [50e]

b 43

Nao encontrado 43.

b 35

Encontrado 35

e

35 40 45 50

f

Ao imprimir, exiba os itens separados por uma tabulação. Ao exibir em largura, você deve começar cada linha com [n], onde n é o nível atual. Depois, exiba os valores do nível no formato (COR)CHAVE<ESPAÇO>[CHAVEPAI e/d], onde e/d indica se é filho esquerdo ou direito. Cada nodo impresso deve estar separado por uma tabulação.

A saída do seu programa deve seguir exatamente o modelo disposto neste documento, sob pena de descontos de nota ou até anulação da sua prova.

6 Arquivos a serem entregues

Os arquivos devem estar em um diretório de nome *antesprova2_GRR*. Se, por exemplo, o grr do aluno é 1234, o diretório contendo os arquivos deve se chamar *antesprova2_1234*. Compacte esse diretório, sendo que a versão compactada vai se chamar *antesprova2_1234.tar.gz*. Ao descompactar o arquivo *antesprova2_1234.tar.gz*, deverá ser criado um diretório de nome *antesprova2_1234* que conterá todos os demais arquivos. O diretório deve conter o seguinte:

- arquivos de código fonte .c e .h do programa;
- makefile;

Não inclua quaisquer outros arquivos irrelevantes para o projeto. Por exemplo, não inclua binários compilados, ou arquivos objeto (.o). A inclusão de arquivos irrelevantes pode acarretar em descontos de nota.

7 Entrega

Submeta os arquivos na UFPRVirtual até o dia da prova, quando os arquivos serão utilizados por você.

8 Distribuição da Nota

Alguns itens que serão levados em consideração no dia da prova, considerando uma nota entre 0 e 100 pontos:

- Plágio ou uso de IA Generativa: perda total da pontuação para todos os envolvidos. Isso é válido mesmo para casos onde o plágio se refere a apenas um trecho do código;
- Entrega, mas não comparecimento no dia da prova, ou incapacidade de implementar os itens solicitados durante a prova, acarreta na perda total dos pontos.

- Algoritmos implementados em linguagem diferente de C serão desconsiderados;
- Não submissão via UFPRVirtual acarreta na perda total dos pontos;
- Falta de algum arquivo requisitado: desconto de 10 a 100 pontos;
- Inclusão de arquivos desnecessários (lixo): desconto de 5 a 20 pontos;
- Erros e avisos de compilação: desconto de 5 a 100 pontos;
- Nomes de arquivo incorretos: pelo menos 10 pontos por arquivo;
- Arquivo com formato incorreto: 10 a 100 pontos;
- Problemas de programação, como uso de funções descontinuadas, programação insegura, e outros problemas: 10 a 100 pontos;
- Corrupção, vazamentos de memória, *user after free*, *double free*, entre outros problemas de gerenciamento de memória: 10 a 100 pontos;
- Formato de saída incorreto: 10 a 100 pontos;
- Algoritmos implementados incompatíveis com uma árvore Red-Black serão desconsiderados;

9 Demais Regras

- Dúvidas ou casos não especificados neste documento podem ser discutidos com o professor até a **data de entrega**. **Não** serão aceitas reclamações após a data da entrega.
- Os alunos podem (e devem) procurar o professor para tirar dúvidas até a data da prova.
- O descumprimento das regras dispostas nesse documento podem acarretar na perda parcial ou total da nota.