

Árvore Rubro-Negra

SERGIO SIVONEI DE SANT'ANA FILHO GRR20242337
EDUARDO KALUF GRR20241770

April 29, 2025

1 Apresentação

O objetivo deste trabalho é a implementação da árvore Red-Black, também conhecida como Rubro-Negra, utilizando a linguagem C, com base no livro "Algoritmos: Teoria e Prática" do H. Cormen. Esta é uma árvore binária que utiliza um campo de cor a fim de manter a árvore o mais balanceada possível, fazendo com que as operações feitas nela sejam executadas com tempo $\mathcal{O}(\log n)$

2 Implementação

A implementação foi feita com base nas notas de aula e no próprio livro do Cormen, com modificações a fim de cumprir as especificações do trabalho. A sua construção foi feita do zero, sem utilizar funções de uma BST já implementada, focando em deixar o trabalho mais sucinto e didático.

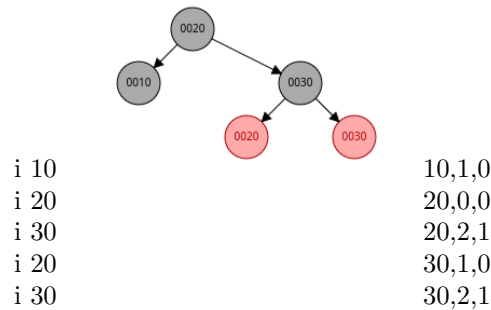
Optamos por usar um sentinela que guarda um ponteiro para o nodo raiz e para o nodo **NIL**. O nodo especial **NIL** é utilizado para se referir a **NULL**, sendo então, um nó preto (na Red-Black, ponteiros para **NULL** são tratados como nós pretos), o qual todos os nodos folhas da árvore apontam.

Como dito anteriormente, modificações foram feitas para atender aos critérios do trabalho, a função "rb delete", por exemplo, quando deleta a raiz, busca o antecessor (nodo com a chave imediatamente menor que a chave da raiz) para se tornar nova raiz da árvore. No header "rb.h" constam informações mais particulares sobre cada função implementada.

O algoritmo de busca utilizado no código é a busca binária recursiva, pois a Rubro-Negra se trata de uma árvore binária específica.

Os comandos aceitos são de inserção e de remoção, representados pelos caracteres "i" e "r" respectivamente, seguindo o formato "instrução valor".

Como dito nas especificações, a saída é expressada em *ordem* na *stdout*, ordenando por valor, nível e cor, respectivamente. Em caso de valores repetidos, segue ordem crescente de nível, pois chaves idênticas são inseridas à direita, portanto impressas depois, seguindo sempre **Esquerda** → **Raiz** → **Direita**.



3 Diretórios e Arquivos

- **rb.h**
Arquivo header que contém a documentação de cada função utilizada.
- **rb.c**
Arquivo de implementação das funções da árvore Red-Black.
- **main.c**
Arquivo responsável por receber os comandos passados pelo terminal e chamar as funções correspondentes, gerenciando o fluxo de dados do trabalho.
- **Makefile**
Arquivo simples utilizado para compilar e fazer a linkagem do trabalho.
 - **make** → gera o executável **myrb**
 - **make clean** → remove os arquivos gerados pela compilação
- **testes e testes.sh**
testes.sh é um script simples que verifica, através dos testes e soluções armazenados no diretório **testes**, se a árvore implementada está apresentando o comportamento esperado.
O script compila automaticamente o projeto e o executa utilizando o comando **diff**. Sendo assim, caso nada seja impresso no terminal, todos os testes foram bem-sucedidos.

4 Conclusão

Ao longo deste trabalho, foi possível compreender de forma mais aprofundada como uma árvore Red-Black pode ser implementada, analisando sua estrutura, funcionamento e as vantagens que oferece. Foi possível ver que esse tipo de árvore proporciona uma eficiência ótima nas operações de inserção, remoção e busca. Além disso, sua praticidade de uso em diversas aplicações, como no próprio sistema de gerenciamento de arquivos do linux, reforça sua importância como estrutura de dados.