



Instituto Tecnológico de Aeronáutica
Divisão de Ciência da Computação

CTC-12

Laboratório 02

Balanced Tree

Aluno:

Daniel Araujo Cavassani

Professor:

Luiz Gustavo Bizarro Mirisola

1) Descrição da estrutura escolhida

Estrutura utilizada: **RedBlack Tree**.

Essa estrutura é uma estrutura de dados do tipo árvore binária de busca balanceada, onde cada nó possui uma cor (vermelho ou preto) e segue algumas propriedades específicas para manter o balanceamento durante as operações de inserção e remoção.

Referência utilizada: livro “Introduction to Algorithms” (também conhecido como Cormen). A implementação da árvore foi baseada no pseudo-código e explicações deste livro, que apresenta detalhadamente os conceitos básicos da estrutura, incluindo o funcionamento dos nós vermelho-preto e dos nós NIL, além de demonstrar a correção do algoritmo usando as invariâncias do loop e fornecer as operações de rotação e inserção necessárias para manter as propriedades da árvore.

Durante a implementação do código, a estrutura principal seguiu o padrão usual de uma árvore vermelho-preto. No entanto, algumas considerações podem ser úteis para facilitar o entendimento do código:

- Utilizei duas classes, ‘Node’ e ‘RedBlackTree’. A classe ‘Node’ representa os nós individuais da árvore, contendo informações como chave (key), valor (value), cor e ponteiros para o nó pai, nó filho à esquerda e nó filho à direita. Já a classe ‘RedBlackTree’ gerencia a árvore em si, contendo um ponteiro para a raiz, um ponteiro para um nó especial NIL, e métodos para operações como adicionar elementos, buscar elementos em um intervalo e rotacionar nós e excluir sub-árvores.
- No código, foi utilizado um nó NIL especial, representado por um objeto da classe ‘Node’. Este nó funciona como uma sentinela para indicar os limites da árvore, simplificando as operações na árvore e evitando a necessidade de verificar explicitamente se um nó é nulo.
- Em vez de criar múltiplos nós NIL, optou-se por criar um único nó NIL e fazer todos os nós folha apontarem para ele. Isso reduz a quantidade de memória utilizada pela estrutura.
- Optou-se por utilizar o método ‘createNode’ para criar nós na árvore, já configurando o nó com os valores de chave, valor e cor apontando para o nó NIL. Isso facilita a criação de novos nós e simplifica o código em várias partes.

2) Demonstrativo

Temos a seguir as comparações feitas, mostrando que ambos os algoritmos (o que eu fiz e o que já é implementado na `stl::multimap`) são consistentes em termos de complexidade temporal. Nota-se, também, que minha implementação apresenta uma grande quantidade de ruídos! Temos os gráficos associados a essa afirmação:

