Implementação de índice de árvore B+ e hash

Beatriz Rodrigues de Oliveira Paiva
Instituto Federal de Minas Gerais (IFMG) – Campus Bambuí
beatrizrodrigues2_@hotmail.com

RESUMO

Este artigo apresenta uma implementação em Python da Árvore B+ e de um índice de hash, que são estruturas de dados utilizada para a organização e busca de valores em grandes quantidades de dados. A implementação da árvore foi desenvolvida utilizando a técnica de programação orientada a objetos e inclui funções para inserção, remoção, busca por igualdade, busca por intervalo e verificação de existência de um valor na árvore. Já a implementação do hash tem a função de inserir pares chave-valor, procurar chaves, remover chaves e atualizar a tabela hash quando ocorrer estouro.

Palavras-chave: Árvore B+, Hash, Python, Busca por igualdade, Busca por intervalo.

1 INTRODUÇÃO

Estruturas de dados são fundamentais em programação, permitindo a organização e armazenamento de informações de maneira eficiente e útil. Duas estruturas de dados importantes são árvores e hashes.

Um código de hash é uma representação única e compacta de um conjunto de dados. Eles são amplamente utilizados na segurança da informação, criptografia, armazenamento de dados e na verificação de integridade de arquivos. Um código de hash é gerado a partir de uma entrada de dados, e uma pequena mudança na entrada resulta em uma saída completamente diferente. Dessa forma, é possível verificar se o conteúdo de um arquivo foi alterado sem a necessidade de armazenar o conteúdo completo. Neste artigo, exploraremos a teoria por trás dos códigos de hash e sua aplicação na segurança da informação.

Uma árvore é uma estrutura de dados hierárquica, composta por nós e arestas. Cada nó pode ter vários filhos, e o nó raiz é o ponto de partida da árvore. As árvores são utilizadas em muitas aplicações, como sistemas de arquivos, organização de categorias e na representação de expressões matemáticas.

A Árvore B+ é uma estrutura de dados que permite armazenar e recuperar informações de forma eficiente. Ela é uma variante da Árvore B e é amplamente utilizada em sistemas de banco de dados relacionais, em especial em sistemas de armazenamento de dados grandes e complexos.

Por outro lado, a Tabela Hash é uma estrutura de dados que usa funções matemáticas para mapear informações para posições únicas em um vetor. Isso permite uma busca e armazenamento

de dados mais rápidos, uma vez que as informações são acessadas diretamente pelo seu endereço na tabela. No entanto, a tabela hash pode sofrer com colisões, o que significa que mais de uma informação pode ser mapeada para a mesma posição.

2 METODOLOGIA OU MATERIAL E MÉTODO

Uma árvore B+ foi utilizado uma classe Arvore representa a estrutura de uma árvore B+. O objetivo da árvore é armazenar valores de forma ordenada. O método init instância a árvore, e especifica o grau da árvore. O método inserir é responsável por inserir um valor na árvore, enquanto o método remove é utilizado para remover um valor da árvore.

O método buscar é utilizado para buscar um valor na árvore. O método buscar_por_igualdade é utilizado para buscar uma chave específica na árvore, e o método buscar_intervalo é utilizado para buscar valores dentro de um determinado intervalo. Todos os métodos retornam informações sobre a localização do valor na árvore.

3 RESULTADOS E DISCUSSÃO

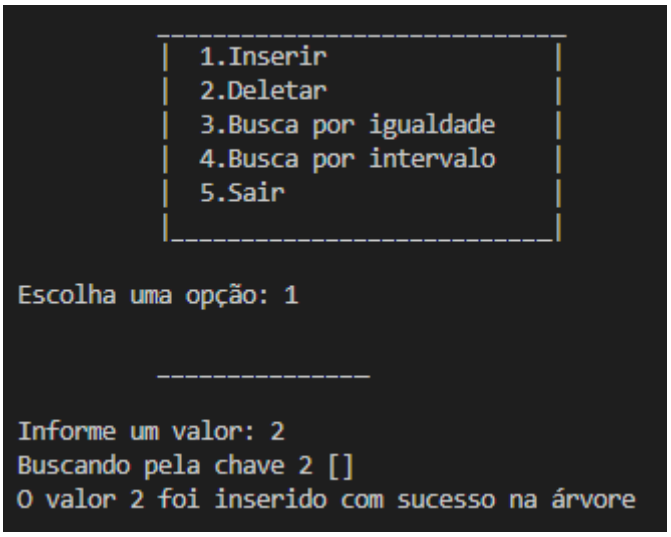
A busca por igualdade e a busca por intervalo são implementadas de forma eficiente, permitindo que sejam realizadas de forma rápida. A busca por igualdade utiliza o método de busca por igualdade, procurando pelo valor específico na árvore, retornando a página e a posição da chave se encontrada. Já a busca por intervalo utiliza o método de busca por intervalo, retornando uma lista com todos os valores encontrados dentro do intervalo especificado.

A inserção e a remoção são implementadas de forma que sejam realizadas de forma balanceada, mantendo a árvore sempre balanceada. Em caso de inserção, o valor é inserido na folha correta, e caso haja necessidade, a folha é dividida, mantendo a árvore balanceada. Já na remoção, o valor é removido da folha correta, e caso haja necessidade, a folha é fundida ou o valor é redistribuído, mantendo a árvore balanceada.

Em resumo, a implementação da árvore B+ permite que sejam realizadas operações de busca, inserção e remoção de forma eficiente, garantindo sempre o balanceamento da árvore. O grau da árvore influencia na sua performance, sendo importante que seja escolhido de forma adequada, considerando as operações que serão realizadas na árvore.

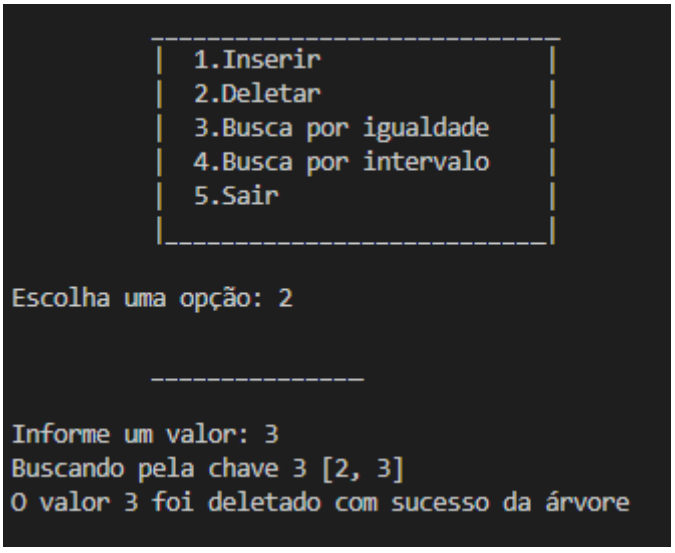
Logo abaixo está o resultado da inserção, remoção, busca por igualdade e busca por intervalo realizado neste trabalho:

Figura 1 - Inserindo.



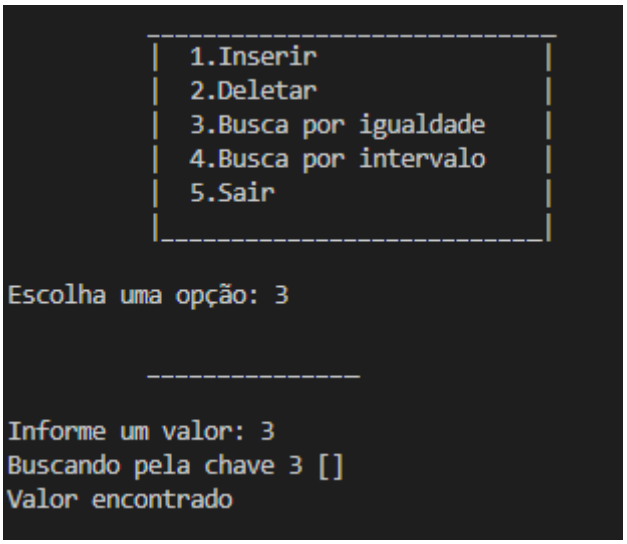
Fonte: Autora

Figura 2 - Deletando.



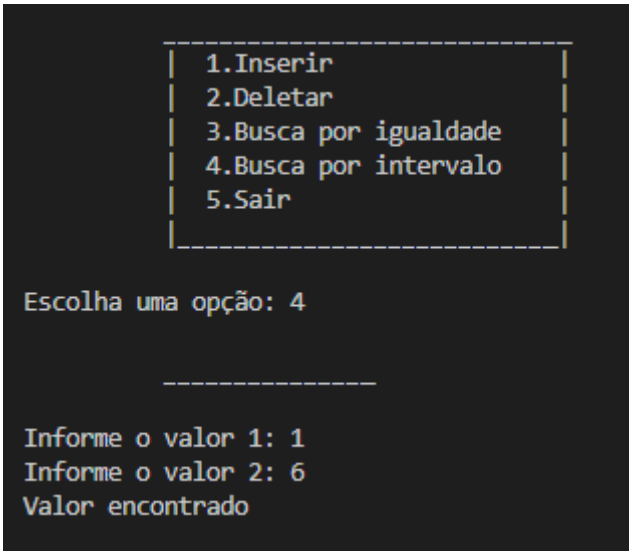
Fonte: Autora

Figura 3 - Busca por igualdade.



Fonte: Autora

Figura 4 - Busca por intervalo.



Fonte: Autora

4 CONCLUSÃO

A implementação de uma Árvore B+ em Python é uma forma eficiente de armazenar informações indexadas. A Árvore B+ é uma estrutura de dados de árvore que possui uma série de

vantagens sobre outras estruturas de dados, como a Árvore B original. A implementação desta estrutura de dados é simples e pode ser facilmente adaptada para diferentes aplicações.