|  |  |  |
| --- | --- | --- |
| **Estudo de caso computacional** | | |
| **Título** | **Árvore binária de busca (BST)** | |
| **Prof. Caio Eduardo** | **Turma:** |  |
|  | | |
| **Nome** |  | |
| **Nome** |  | |
| **Nome** |  | |

**Objetivos**

* **Objetivo Geral**:

Ensinar os alunos como utilizar uma árvore binária de busca (BST) para armazenar e organizar produtos com base em uma chave única, como o ID do produto.

* **Objetivos Específicos**:

Entender o conceito de árvore binária de busca (BST).

Implementar a estrutura de dados BST para produtos, onde cada produto é representado por um dicionário com informações como ID, nome, preço e estoque.

Explorar como inserir, buscar e listar produtos utilizando a árvore binária.

Aplicar conceitos de eficiência de algoritmos em uma estrutura de dados realista.

**Materiais e Métodos**

**Materiais:**

* Computadores com Python instalado.
* Editor de código (Visual Studio Code, PyCharm ou similar).
* Acesso a um terminal ou prompt de comando.

**Métodos:**

* A aula será prática, com os alunos implementando a estrutura de dados em Python.
* A implementação base é fornecida apenas com a definição de uma árvore binária de busca (sem a lógica de produtos).
* O código será incrementado ao longo da aula, onde os alunos adicionarão as funcionalidades necessárias para lidar com os produtos.
* Essa atividade deve ser realizada em grupo de 3 alunos, no máximo.

**Procedimentos**

Introdução:

* Explicação do conceito de árvore binária de busca (BST).
* Apresentação do caso de uso prático onde uma BST pode ser aplicado.

Apresentação da Estrutura do Código base:

Fonte: <https://github.com/caio-ireno/Estudo-de-Caso-Computacional---Estrutura-de-Dados>

* Apresentação da implementação da árvore binária de busca (BST) com as classes Node e BST.
* Mostrar que a classe **Node** representa cada nó da árvore com um valor e ponteiros para os filhos esquerdo e direito.
* Explicar como o método **insert** funciona para inserir um novo produto na árvore, levando em conta a ordem do ID do produto.

Atividade 1: Implementação da Estrutura de Dados:

* Faça a implementação de árvore binária de busca (BST) para armazenar objetos de produto (representados por um dicionário, contendo ID, nome, preço e estoque).
* Criar a classe **ProductInterface** para representar os produtos.
* Cada produto será inserido na árvore com base no seu **product\_id**.

Atividade 2: Implementação da Busca e Percurso em Ordem:

* Após a implementarem a inserção, oriente-os a criar a funcionalidade de busca na árvore, que deve retornar um produto pelo seu ID.
* Ensine como implementar o percurso em ordem para listar todos os produtos de forma ordenada com base nos seus IDs.

Atividade 3: Testes e Validação:

* Os alunos devem testar suas implementações inserindo produtos, realizando buscas e verificando o percurso da árvore.
* Discuta como os testes podem ser feitos, por exemplo, buscando por um produto pelo ID ou verificando se a árvore retorna os produtos na ordem correta.

**Discussão Final:**

* Faça uma reflexão sobre a aplicação de árvores binárias de busca na realidade (exemplo: gerenciar um estoque de produtos).
* Discuta possíveis melhorias na implementação, como balanceamento de árvores, e outros algoritmos de busca.
* Responda as questões, justificando e discutindo com o grupo.

**Atividades**

**Implementação da Estrutura de Dados:**

Crie a classe **ProductInterface** que armazena as informações de um produto de acordo com a tabela abaixo.

|  |  |
| --- | --- |
| **Informação** | **Descrição** |
| **ID** | Número de identificação único do produto. |
| **Nome** | Nome do produto |
| **Preço** | Preço do Produto |
| **Estoque** | Quantidade de produtos disponíveis no estoque |

Implemente a classe **Node** e **BST** para construir a árvore binária de busca.

Implemente a função de inserção de produtos na árvore binária (com base no ID do produto).

**Busca na Árvore:**

Adicione a funcionalidade de busca na árvore para encontrar um produto pelo **product\_id.**

**Listagem dos Produtos (Percurso em Ordem):**

Implemente o percurso em ordem da árvore, que retorna os produtos ordenados pelo **product\_id.**

**Testes e Validação:**

Teste a inserção e a busca de produtos na árvore.

Implemente um pequeno script para validar se a árvore está funcionando corretamente (inserção, busca e listagem).

**Perguntas**

1. Explique por que a busca em uma árvore binária de busca (BST) é mais eficiente que a busca em uma lista desordenada. Baseie sua resposta na diferença entre os tempos de complexidade O(n) e O(log n).

2. Considere a seguinte afirmação:

"As subárvores esquerda e direita de qualquer nó em uma BST também são árvores binárias de busca."

Por que essa propriedade é importante para garantir a eficiência da estrutura?

3. Explique para que tipo de problema os percursos abaixam seriam uteis, de um exemplo de um caso real.

a) Em Ordem.

b) Pré-Ordem.

c) Pós-Ordem.

4. Quais são as principais vantagens de utilizar uma BST para organizar dados?