	Cursos de Informática	2024/1 Trabalho 1 GA (1,0) Est. de Dados Avançadas
	Prof. Gilberto Irajá Müller	


Contexto: É inquestionável a aplicação de estruturas do tipo árvore em diversas áreas conforme visto em aula. Pensando nisso, você deverá desenvolver alguns métodos baseados na árvore de pesquisa binária apresentada em aula.

Classe abstrata: use a classe abstrata a seguir e isso significa que não deverá ser alterado nenhum nome de método. Lembre-se que, para a validação de cada método, o professor utiliza testes automatizados e, portanto, é fundamental que não seja alterado nada.

Passo 1). Crie a seguinte classe abstrata e vincule-a à classe *BinarySearchTree* apresentada em aula.

```
class BinarySearchTreeADT(ABC):
    @abstractmethod
    def clear(self) -> None: ...
    @abstractmethod
    def is_empty(self) -> bool: ...
    @abstractmethod
    def search(self, key: object) -> object: ...
    @abstractmethod
    def insert(self, key: object, value: object) -> None: ...
    @abstractmethod
    def delete(self, key: object) -> bool: ...
    @abstractmethod
    def pre_order_traversal(self) -> None: ...
    @abstractmethod
    def in_order_traversal(self) -> None: ...
    @abstractmethod
    def post_order_traversal(self) -> None: ...
    @abstractmethod
    def level_order_traversal(self) -> None: ...

    # Métodos para desenvolver
    @abstractmethod
    def size(self) -> int: ...
    @abstractmethod
    def degree(self, key: object) -> int: ...
    @abstractmethod
    def height(self, key: object) -> int: ...
    @abstractmethod
    def depth(self, key: object) -> int: ...
    @abstractmethod
    def descendent(self, key: object) -> str: ...
```


	Cursos de Informática	2024/1 Trabalho 1 GA (1,0) Est. de Dados Avançadas
	Prof. Gilberto Irajá Müller	

Passo 2). Desenvolva em Python cada método conforme regra de negócio descrita no Quadro 1. **Lembre-se que poderá ser criado qualquer método que facilite o desenvolvimento.**

Quadro 1: Descrição dos Contratos

Contrato	Descrição
size	Objetivo: retornar a quantidade de nós da árvore.
	Parâmetros: nenhum.
	Retorno: quantidade de nós da árvore ou zero caso a árvore esteja vazia.
degree	Objetivo: retornar o grau de um nó.
	Parâmetros: key: chave que se deseja verificar o grau.
	Retorno: grau do nó que é representado pela chave. Caso não encontrada a chave, retornar -1.
height	Objetivo: retornar a altura de um nó.
	Parâmetros: key: chave que se deseja verificar a altura.
	Retorno: altura do nó que é representada pela chave. Caso não encontrada a chave, retornar -1.
depth	Objetivo: retornar a profundidade de um nó.
	Parâmetros: key: chave que se deseja descobrir a profundidade.
	Retorno: a profundidade do nó representada pela chave. Caso não seja encontrada a chave, retornar -1.
descendent	Objetivo: retornar os descendentes lado a lado.
	Parâmetros: key: chave que se deseja descobrir os descendentes.
	Retorno: lista em texto com as chaves lado a lado (separadas por espaço) que representam os descendentes. Caso não seja encontrada a chave, retornar None.

Fonte: elaborado pelo autor.

	Cursos de Informática	2024/1 Trabalho 1 GA (1,0) Est. de Dados Avançadas
	Prof. Gilberto Irajá Müller	

Avaliação

- Todos os métodos devem ser desenvolvidos de forma RECURSIVA. Caso algum método não seja recursivo, ficará com a nota zerada, mesmo sendo funcional;
- Código o mais claro possível;
- Não será aceito a entrega (via Moodle) do trabalho após a data limite;
- Postar apenas a classe *BinarySearchTree*;
- Colocar no cabeçalho da classe *BinarySearchTree* o número do grupo, além do nome completo dos integrantes;
- Não será aceito trabalho igual ao de outros colegas. Esta prática é chamada de plágio.