Equipe: Davi Sousa Soares
Victor Kauan da Silva Miranda
João Pedro Saleh de Sousa

Professor: Raimundo Santos Moura

DEZEMBRO
2024

Conteúdo

1 Introdução

Árvore binária é uma estrutura de dados abstrata onde cada elemento aponta para dois nós chamados de nós filhos e possui um nó que aponta para ele chamado de nó pai,com exceção da raiz que é o primeiro elemento da árvore e por isso não possui um nó pai.

1.1 Objetivos

O objetivo desse trabalho é apresentar uma interface mostrando a execução de um exemplo de árvore binária além de fornecer informações a respeito da mesma, como altura, comprimento interno e etc

---

1.2 Especificações

A interface e a estrutura da árvore foram feitas usando python. foi usada a biblioteca tkinter para a criação da interface e programação orientada a objetos para a criação da árvore, a interface se ocupa apenas de mostrar os resultados do código da árvore.

---

2 Funcionalidades

Este tópico é destinado a explicar as capacidades do código realizado

---

2.1 Árvore Binária em python

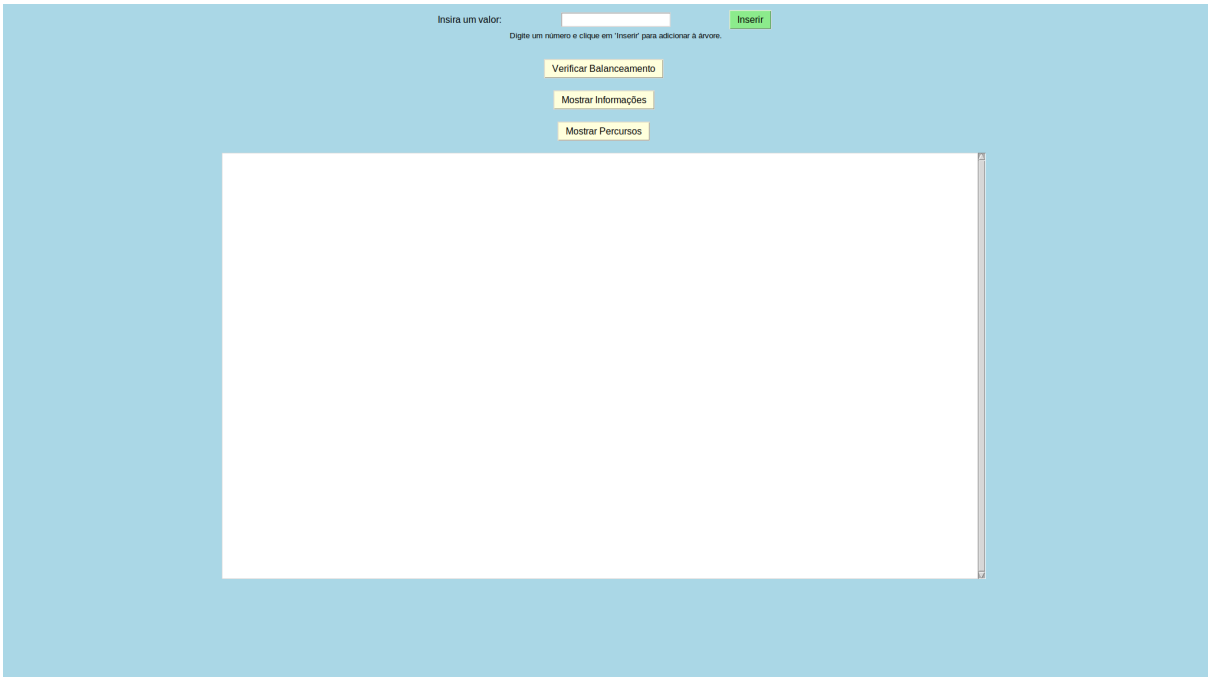
A Árvores possui um método de inserção que decide se o elemento irá para a esquerda ou para a direita além de vários métodos para percorrer a árvore (pós-ordem, pré-ordem, em-ordem e por largura)

```
def inOrder(node, elems = []):
   if node.left is not None: inOrder(node.left, elems)
   elems.append(node.key)
   if node.right is not None: inOrder(node.right, elems)

def postOrder(node, elems = []):
   if node.left is not None: inOrder(node.left, elems)
   if node.right is not None: inOrder(node.right, elems)
   elems.append(node.key)

def preOrder(node, elems = []):
   elems.append(node.key)
   if node.left is not None: inOrder(node.left, elems)
   if node.right is not None: inOrder(node.right, elems)
```

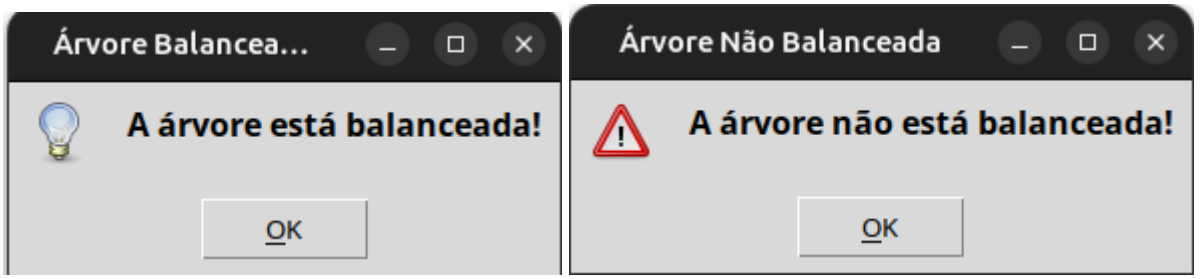
cada objeto nó possui métodos que ajudam na interface como por exemplo sua altura na árvore

```
class Node:

   def __init__(self, val):
       self.key = val
       self.left = None
       self.right = None
   def height(self):

       if not self:
           return 0
       left_height = self.left.height() if self.left else 0
       right_height = self.right.height() if self.right else 0
       return 1+max(left_height, right_height)

   def fb(self):
       return -height(self.left) + height(self.right)
```

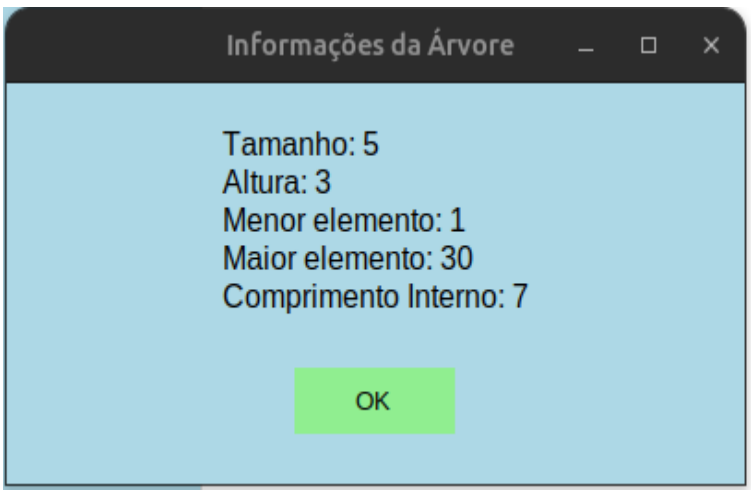
2.2 Interface

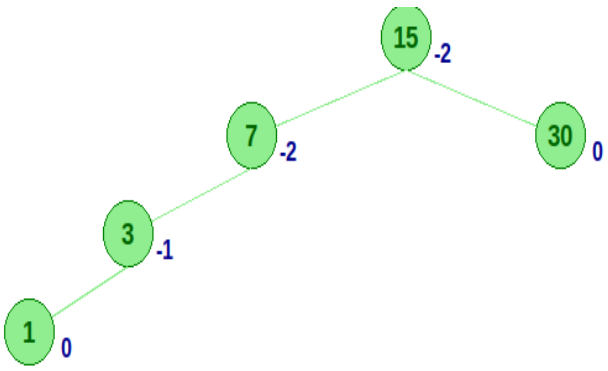


Página inicial da interface

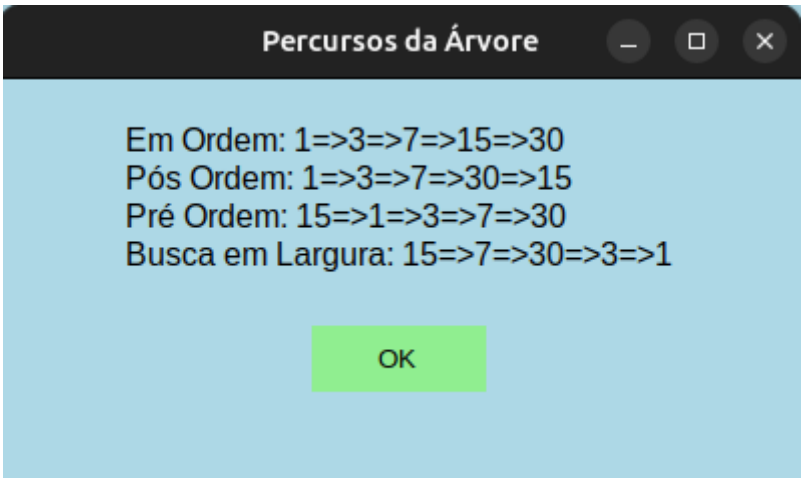


Possíveis respostas ao clicar no botão “verificar balanceamento”





exemplo de resultado ao clicar em “mostrar informações”



exemplo de resultado ao clicar em “mostrar percurso” usando o mesmo exemplo anterior