



PRACTICA 2

Mercado Manzo Luis Alfonso



15 DE OCTUBRE DE 2023

```

import tkinter as tk
from tkinter import ttk
import networkx as nx
import matplotlib.pyplot as plt

class TreeNode:
    def __init__(self, key):
        self.key = key
        self.left = None
        self.right = None

class BinarySearchTree:
    def __init__(self):
        self.root = None

    def insert(self, key):
        self.root = self._insert(self.root, key)

    def _insert(self, node, key):
        if node is None:
            return TreeNode(key)
        if key < node.key:
            node.left = self._insert(node.left, key)
        elif key > node.key:
            node.right = self._insert(node.right, key)
        return node

    def search(self, key):
        return self._search(self.root, key)

    def _search(self, node, key, path=None):
        if node is None:
            return False, path
        path.append(node.key)
        if key == node.key:
            return True, path
        if key < node.key:
            return self._search(node.left, key, path)
        return self._search(node.right, key, path)

    def visualize_tree(root):
        G = nx.Graph()
        visualize_tree_helper(root, G)
        pos = nx.spring_layout(G, seed=42)
        nx.draw(G, pos, with_labels=True, node_size=500, node_color="red")

```

```

plt.title("Árbol Binario de Búsqueda")
plt.show()

def visualize_tree_helper(node, G):
    if node:
        if node.left:
            G.add_edge(node.key, node.left.key)
            visualize_tree_helper(node.left, G)
        if node.right:
            G.add_edge(node.key, node.right.key)
            visualize_tree_helper(node.right, G)

# Crear el árbol binario con palabras reservadas
bst = BinarySearchTree()
palabras_reservadas = ["if", "else", "for", "while", "if else",
                        "float", "return", "break", "continue", "switch"]
for palabra in palabras_reservadas:
    bst.insert(palabra)

# Visualizar el árbol binario
visualize_tree(bst.root)

# Función para buscar y mostrar el recorrido de búsqueda
def buscar_palabra():
    palabra_buscar = entrada.get()
    encontrado, path = bst.search(palabra_buscar)
    if encontrado:
        resultado.config(text=f"{palabra_buscar} encontrado en el
árbol: {path}")
    else:
        resultado.config(text=f"{palabra_buscar} no encontrado en el
árbol")

# Configuración de la ventana principal
ventana = tk.Tk()
ventana.title("Árbol Binario de Búsqueda")

etiqueta = tk.Label(ventana, text="Palabra a buscar:")
entrada = ttk.Entry(ventana)
boton_buscar = ttk.Button(ventana, text="Buscar",
command=buscar_palabra)
resultado = tk.Label(ventana, text="")

etiqueta.pack()
entrada.pack()

```

```
boton_buscar.pack()  
resultado.pack()  
  
ventana.mainloop()
```

Árbol Binario de Búsqueda

