

Integrantes:

Felipe Alvarado

Sebastián Adriazola

Sebastián Utreras

Gerardo Cerón

Melian Lucabeche

Referencia 1:

Árbol Trinario Recursivo

<https://gist.github.com/alexortizrosado/e852bb3129110538c431>

Detalle: Eliminación de nodos sustituir el nodo a eliminar por el nodo menor del subárbol, pero creaba dos instancias del nodo, en su estado anterior y en su estado al sustituir el nodo
(Solucionado)

Autor: Alex Ortiz-Rosado

Referencia 2-1:

Árbol Trinario Iterativo

Código en js

<https://gist.github.com/siwalikm/3f2084f4f28c6ce851b019b5cd743017>

Autor: Siwalik Mukherjee

Referencia 2-1:

Árbol Trinario Iterativo

<https://chat.openai.com/>

Pregunta:

“programa un árbol trinario con método de agregado y eliminación de manera iterativa en Python”

Fuentes:

1. Documentación oficial de Python: La documentación oficial de Python proporciona información detallada sobre la sintaxis del lenguaje y el uso de las estructuras de datos integradas, como los diccionarios y listas, que son útiles para comprender cómo trabajar con estructuras de datos en Python.
 2. Libros sobre algoritmos y estructuras de datos en Python: Libros como "Problem Solving with Algorithms and Data Structures using Python" de Bradley N. Miller y David L. Ranum proporcionan una cobertura detallada de varios algoritmos y estructuras de datos, incluidos los árboles.
 3. Recursos en línea: Sitios web como GeeksforGeeks, Stack Overflow y Programiz ofrecen tutoriales y ejemplos de código sobre cómo implementar diferentes estructuras de datos en Python, incluidos los árboles.
-