

Reflexión Act 3.4

En esta reflexión, abordaré la experiencia de resolver el problema del "Árbol Abeto". Analizaré lo que aprendí durante el proceso y cómo desarrollé el programa para abordar este desafío.

El primer paso en la resolución de este problema fue comprender completamente lo que se requería. Aprendí que la definición de un "abeto" en el contexto de un árbol enraizado implica que cada vértice no hoja debe tener al menos 3 hijos hojas. Este entendimiento es crucial antes de comenzar a escribir el código, ya que determina la lógica que se debe implementar.

Otro aspecto importante de la resolución de este problema fue el análisis de ejemplos de entrada y salida proporcionados. Estos ejemplos me permitieron tener una idea clara de lo que se esperaba de la solución. A través de este análisis, pude confirmar que debía verificar que el árbol cumpliera con la definición de un "abeto" y proporcionar "Yes" o "No" en consecuencia.

El desarrollo del programa en C++ fue el siguiente paso. Diseñé una lógica que involucraba contar el número de hijos de cada vértice que no es hoja y verificar si cumplía con la definición de un "abeto". Implementé esta lógica utilizando vectores y bucles.

Ejecutar el programa con ejemplos de entrada fue fundamental. Durante esta fase, identifiqué errores en la salida inicial y depuré el código para corregirlos. Fue un recordatorio importante de la necesidad de verificar cuidadosamente los resultados y tener paciencia en la resolución de problemas algorítmicos.

En mi primer intento, cometí errores en la interpretación de la salida para el segundo ejemplo de entrada. Aprendí la importancia de reconocer y corregir errores, y esto me llevó a una revisión más profunda de las definiciones y restricciones del problema. La corrección del código se hizo posible una vez que comprendí mejor el problema.

En resumen, resolver el problema del "Árbol Abeto" fue un ejercicio valioso que me recordó la importancia de una comprensión profunda del problema, la paciencia en la depuración y la iteración en el desarrollo de programas. También subrayó la necesidad de verificar cuidadosamente los resultados, especialmente al abordar problemas algorítmicos. La resolución exitosa de este problema proporcionó una valiosa experiencia en el campo de la programación y la resolución de desafíos computacionales.

Link de Git