Documentación de arbol binario

Antonio Ramírez Juan Elías

Nava Soler Raúl

Jerónimo Rojano Kevin Marzul

Introducción

La implementación de un árbol binario en lenguaje C es un ejercicio clave en el estudio de las estructuras de datos. Este proyecto tiene como objetivo principal gestionar un árbol binario, permitiendo realizar operaciones de inserción, búsqueda y visualización de nodos, así como llevar a cabo recorridos transversales. La motivación detrás de este proyecto se basa en la necesidad de entender y aplicar los conceptos teóricos de los árboles binarios en un contexto de programación práctica. Además, busca demostrar la eficiencia de los árboles binarios en la gestión y recuperación de datos.

Marco de referencia

Teoría de arboles binarios

Los árboles binarios son estructuras de datos muy comunes en el campo de la informática. Cada nodo en un árbol binario puede tener hasta dos hijos, que se conocen como hijo izquierdo e hijo derecho. Estas estructuras son fundamentales para llevar a cabo algoritmos eficientes para buscar, insertar y eliminar datos.

Aplicaciones en la vida real

Los árboles binarios son fundamentales en los sistemas de bases de datos, ya que permiten organizar y acceder a los datos de manera rápida. Los índices en las bases de datos a menudo se implementan mediante variantes de árboles binarios.

En cuanto a los sistemas de archivos, estos también utilizan árboles binarios para gestionar la jerarquía de directorios y archivos, lo que facilita un acceso rápido y eficiente a la información almacenada.

Método y Material Utilizado

El desarrollo del proyecto sigue un enfoque sistemático, comenzando por definir una estructura de datos binaria que represente los nodos del árbol. Cada nodo tiene campos para almacenar el valor izquierdo, el valor derecho, el índice del padre, el índice del nodo y el nivel del nodo en el árbol. Luego, se implementan funciones para

llevar a cabo las operaciones básicas del árbol: insertar valores, buscar valores, visualizar nodos y realizar recorridos transversales. Estas funciones se integran en un programa principal que interactúa con el usuario a través de un menú de opciones.

Para llevar a cabo este proyecto, se utilizan las siguientes herramientas y recursos:

- **Lenguaje de programación:** C
- **Compilador:** gcc.exe (tdm64-1) 10.3.0 Copyright (C) 2020 Free Software Foundation, Inc. Esto es software libre; vea el c digo para las condiciones de copia. NO hay garantía; ni siquiera para MERCANTIBILIDAD o IDONEIDAD PARA UN PROPÓSITO EN PARTICULAR
- **Entorno de desarrollo:** Visual Studio Code

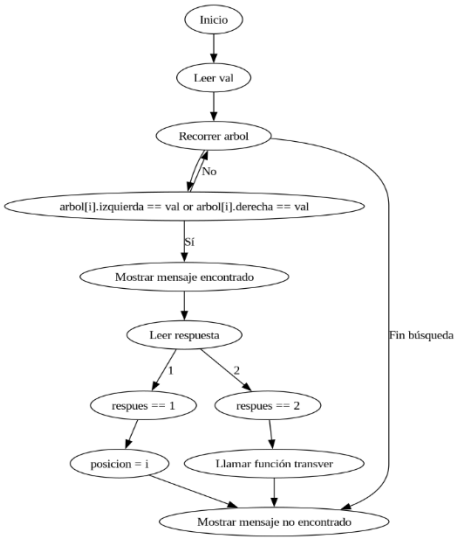
Descripción

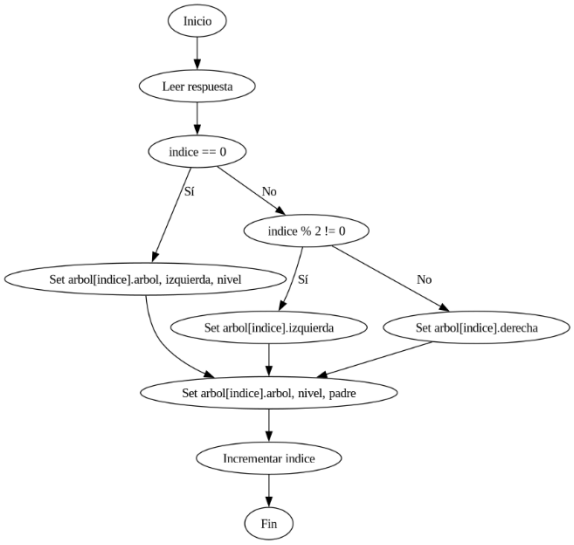
Estructura binaria

La estructura binaria define un nodo del árbol binario, incluyendo campos para almacenar el valor de la izquierda, el valor de la derecha, el índice del padre, el índice del nodo y el nivel del nodo.

Función buscar

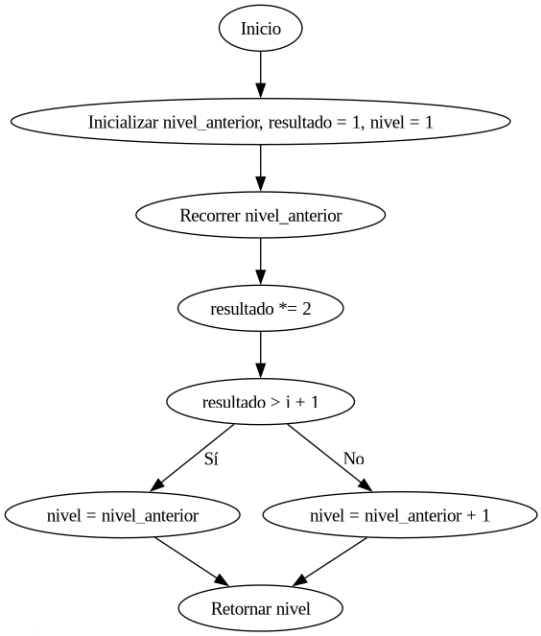
La función buscar permite localizar un valor específico en el árbol binario. Al encontrar el valor, la función imprime su posición, nivel y padre. Esta función es esencial para verificar la existencia de valores en el árbol y obtener información detallada de los nodos correspondientes.





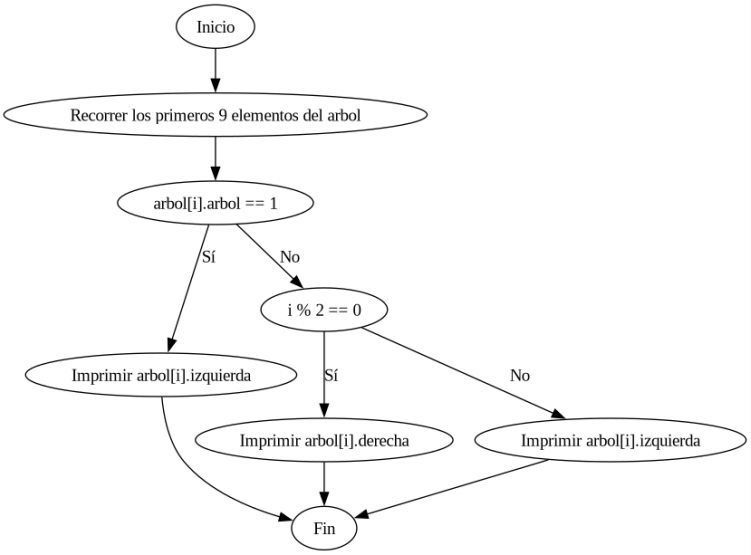
Función insertar_valor

La función insertar_valor se encarga de agregar nuevos valores al árbol binario. Esta función actualiza los atributos del nodo, como el valor de la izquierda o derecha, el índice del padre y el nivel del nodo. La inserción de valores es una operación fundamental que permite construir y expandir el árbol binario.



Función nivel

La función nivel calcula y asigna el nivel de un nodo en el árbol binario. El nivel de un nodo es un indicador de su profundidad en el árbol, comenzando desde 1 para la raíz del árbol. Esta información es crucial para organizar y estructurar correctamente los nodos en el árbol.

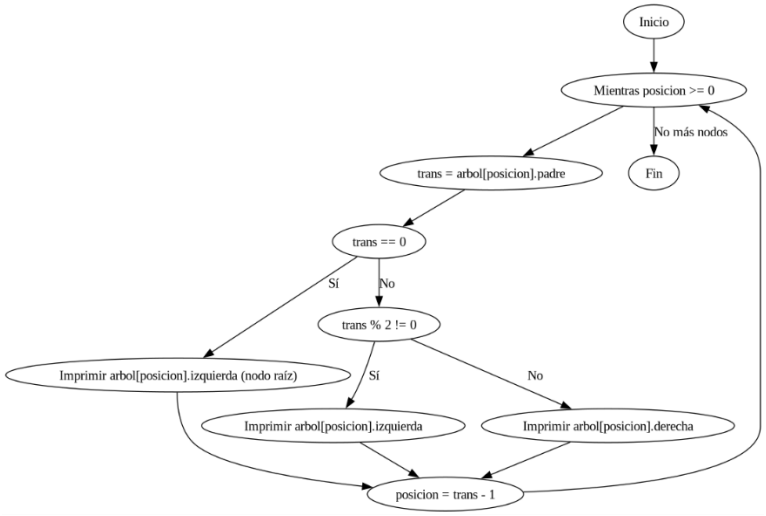


Función mostrar

La función mostrar imprime los primeros nueve nodos del árbol binario, indicando su valor, posición, padre y nivel. Esta función proporciona una representación visual del contenido del árbol, facilitando la comprensión de su estructura y organización.

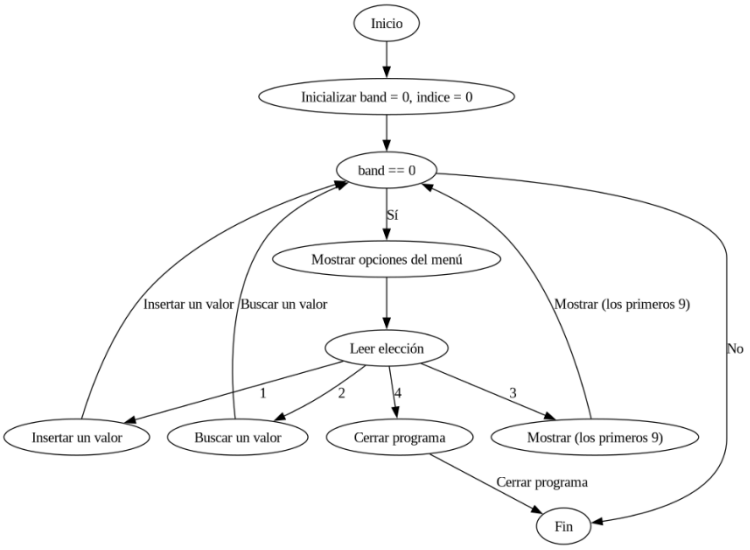
Función transver

La función transver realiza un recorrido transversal del árbol binario a partir de una posición dada, imprimiendo los detalles de cada nodo en el camino. Este recorrido permite obtener una vista completa de los nodos desde el punto de partida hasta la raíz del árbol.



Función main

La función main es la encargada de gestionar la interacción con el usuario. A través de un menú de opciones, permite al usuario elegir entre insertar un valor, buscar un valor, mostrar los primeros nueve nodos o cerrar el programa. Esta función orquesta el flujo de ejecución del programa, llamando a las funciones correspondientes según las elecciones del usuario.



Resultados

Al implementar y ejecutar el programa de gestión de árboles binarios, se esperaba que las funciones de inserción, búsqueda y visualización funcionaran de manera correcta y eficiente. En particular, los resultados esperados incluían:

Inserción de valores: Los valores ingresados debían ser añadidos correctamente al árbol binario, siguiendo las reglas de ubicación de nodos (izquierda para valores menores y derecha para valores mayores).

Búsqueda de valores: Debía ser posible localizar cualquier valor que se hubiera insertado previamente en el árbol y obtener información sobre su posición y nivel dentro de este.

Visualización de nodos: La función de visualización debía mostrar adecuadamente los primeros nueve nodos del árbol, incluyendo sus valores, posición, padre y nivel.

Recorrido transversal: Se debería poder realizar un recorrido transversal del árbol a partir de cualquier nodo dado, mostrando los detalles de cada nodo encontrado en el camino.

Los resultados obtenidos tras la ejecución del programa confirmaron que las funciones operaron como se esperaba. Las pruebas realizadas demostraron que el árbol binario se comportó de manera coherente con las expectativas teóricas.

```
1. Insertar un valor
2. Buscar un valor
3. Mostrar (los primeros 9)
4. Cerrar programa
```

```
1. Insertar un valor
2. Buscar un valor
3. Mostrar (los primeros 9)
4. Cerrar programa
1
¿Cuál número deseas agregar al árbol?
10
```

```
1. Insertar un valor
2. Buscar un valor
3. Mostrar (los primeros 9)
4. Cerrar programa
3
#10 en el nodo 1, su padre es 0 y esta en el nivel 1
#8 en el nodo 2, su padre es 1 y esta en el nivel 2
#0 en el nodo 0, su padre es 0 y esta en el nivel 0
#0 en el nodo 0, su padre es 0 y esta en el nivel 0
#0 en el nodo 0, su padre es 0 y esta en el nivel 0
#0 en el nodo 0, su padre es 0 y esta en el nivel 0
#0 en el nodo 0, su padre es 0 y esta en el nivel 0
#0 en el nodo 0, su padre es 0 y esta en el nivel 0
#0 en el nodo 0, su padre es 0 y esta en el nivel 0
```

```
Ingresa el valor a buscar >> 8

El valor    8 fue encontrado en el nivel  2
1--> SALIR
2-->Recorrido transversal
---> Cualquier otro para continuar la busqueda
>>> 2
#8 en el nodo 2, su padre es 1 y esta en el nivel 2
#10 en el nodo 1, y es el nodo raiz
```

Referencias

Joyanes Aguilar, L. (2005). *Programación en C: metodología, algoritmos y estructura de datos:* (2 ed.). McGraw-Hill España. https://elibro.net/es/lc/bibliotecauv/titulos/50302