



Tecnológico Nacional de México.

Instituto tecnológico de Nuevo
León.



Ingeniería en Sistemas Computacionales

“Reporte Árbol Binario”

Reporte presentado por:

García Hernández Julio César.

#15480089

Guadalupe, Nuevo León a 16 de Febrero del 2018,

INTRODUCCIÓN

Un árbol binario es aquel que solo tiene dos descendientes directos. Los árboles se caracterizan por almacenar sus nodos en forma jerárquica y no en forma lineal como lo son otro tipo de estructuras. Las aplicaciones de los arboles binarios son muy variadas ya que se les puede utilizar para representar una estructura en la cual es posible tomar decisiones con dos opciones en distintos puntos.

Un árbol binario es una estructura de datos, y como todas las demás ya sean pilas o listas, sirve para organizar datos para facilitar su manipulación, ya sea el ingreso, borrado o búsqueda de estos.

Una de las ventajas de los arboles binarios es su parte de la búsqueda ya que como en muchos algoritmos relacionados a esto se necesita tener la información ordenada, y precisamente con los arboles la información entra de esta forma, y puede ser mostrada en sus diferentes tipos de recorridos ya sea en inorden, postorden o preorden.

En este trabajo utilizare como base un antiguo proyecto que realice en la clase de Estructura de datos, con esto doy a entender que la mayoría del código es mío con algunas partes proporcionadas por la maestra.

DESCRIPCIÓN

Se realizó un árbol binario en el cual, como su nombre lo indica, cada nodo tenía dos hijos ya sean de números enteros, letras y/o operadores matemáticos.

El objetivo de este proyecto era el poder imprimir de los 3 tipos de formas de recorridos una ecuación dada por el usuario.

Utilicé 3 clases: Nodo, ArbolBin, Prueba.

- **Nodo:** en esta clase se encuentra la declaración de los nodos al que la asignación de un valor a cada uno de ellos. Los nodos son: Padre, Hoja Izquierda y Hoja Derecha (también podrían ser llamados hijos).
- **ArbolBin:** aquí se inserta cada nodo al igual que es en donde se encuentran los métodos para los recorridos, ya sea en preorden, inorden o postorden, dependiendo el caso. Los recorridos se realizan dependiendo el grado jerárquico de cada elemento insertado.
- **Prueba:** en esta última clase realice un pequeño menú con un do-while y una serie de casos. Las opciones a elegir son: crear árbol, agregar nodo, imprimir (en esta opción se imprimen los 3 tipos de recorridos) y por ultimo salir.

Pseudocódigo:

1.- Inicio

2.- Entrada Nuevo

3.- Condición partiendo de la raíz. $\text{Nodo} == \text{null}$

True Nuevo pasa a ocupar un lugar

En el árbol.

Else $\text{Nuevo} < \text{Nodo}$

Pasamos al Nodo de la izquierda

Se repite desde el paso 3 partiendo

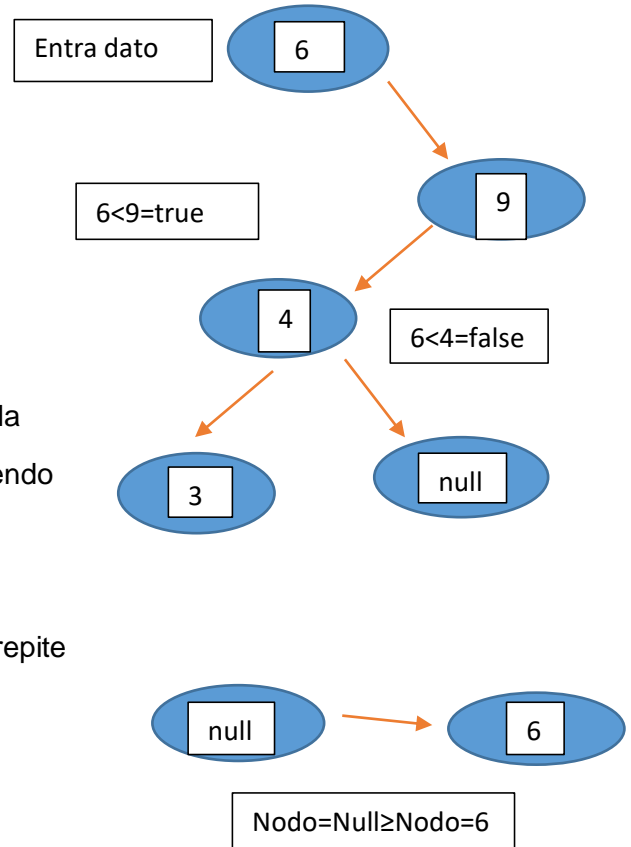
Del nuevo nodo agregado.

En el caso de que sea $\text{Nuevo} > \text{Nodo}$

Se pasa al nodo de la derecha y se repite

El paso 2 desde este nuevo nodo.

4.- fin



RESULTADOS

Con este programa se obtiene en pantalla la impresión de los datos ingresados en preorden, postorden y preorden. Poniendo como ejemplo la siguiente sucesión de números da como resultado lo que a continuación se muestra:

15, 48, 89, 81, 22, 77, 48, 95, 14, 32, 21

Inorden: 14 15 21 22 32 48 48 77 81 89 95

Preorden: 15 14 48 22 21 32 89 81 77 48 95

Postorden: 14 21 32 22 48 77 81 95 89 48 15

CONCLUSIONES:

Con este proyecto se trataba de llegar a la creación de un software capaz de leer una función entera y que esta sea mostrada en los 3 tipos de recorridos, lamentablemente es algo que no logre hacer. No quise basarme en algún ejemplo más complejo sacado de internet porque tal vez utilizan métodos más complejos que no sería capaz de explicar y a su vez de entender, por ende, no quise combinar código más complejo con el programa que ya tenía.

Para concluir, realice o más bien mejore un programa que ya tenía, el cual cumple con su cometido, aunque no con la parte de leer letras y caracteres.