ÁRBOLES BINARIOS ORDENADOS

1. Definición y representación gráfica
2. Inserción
3. Búsqueda
4. Eliminación

SESIÓN **/15**

**INTRODUCCIÓN**

* La clase anterior conocimos los Árboles. Vimos la forma cómo se representan y los términos que se utilizan en esta Estructuras de Datos. Además, vimos un tipo especial de árbol: Los árboles binarios.
* En la presente sesión conoceremos los Árboles Binarios Ordenados, llamados también Árboles Binarios de Búsqueda. Revisaremos sus principales características, su representación gráfica y las operaciones de inserción y búsqueda.
* Culminaremos describiendo los diferentes algoritmos para eliminar un nodo del Árbol Binario Ordenado.

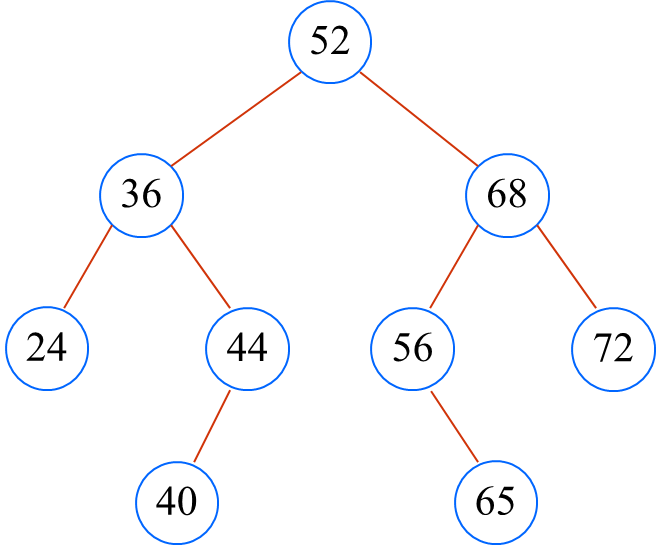
1. **DEFINICIÓN**

Es un Árbol Binario en el cual el valor del sucesor izquierdo de un nodo es menor al valor del nodo; y el valor del sucesor derecho de ese nodo es mayor al valor de dicho nodo.

Se le conoce también como Árbol Binario de Búsqueda.

1. **REPRESENTACIÓN GRÁFICA**

***EJEMPLO:***



***Si tomamos como referencia el 36, vemos que su sucesor izquierdo tiene un valor menor, mientras que su sucesor derecho tiene un valor mayor.***

***Si tomamos como referencia el 68, vemos que su sucesor izquierdo tiene un valor menor, mientras que su sucesor derecho tiene un valor mayor.***

1. **INSERCIÓN**

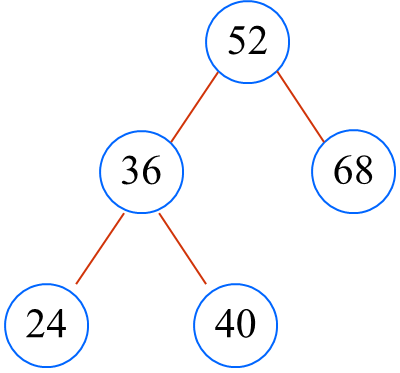
Consiste en agregar un nodo al Árbol. Luego de la inserción, el Árbol debe continuar ordenado.

El algoritmo para insertar un nodo en un Árbol Binario Ordenado, es el siguiente:

1. Crear un nodo.
2. Asignar el valor para el campo info.
3. Asignar el valor null a los campos izq y der.
4. Si el árbol está vacío, el nodo creado será la raíz del árbol y se termina el algoritmo.
5. Repetir los pasos 6, 7 y 8 hasta encontrar la ubicación para el nuevo nodo.
6. Comparar el campo info del nodo creado con el campo info del nodo del árbol.
7. Si el campo info del nodo creado es menor, entonces ir hacia la izquierda.
8. Si el campo info del nodo creado es mayor, entonces ir hacia la derecha.

***EJEMPLO:***

***En el siguient Árbol, inserta el 38.***

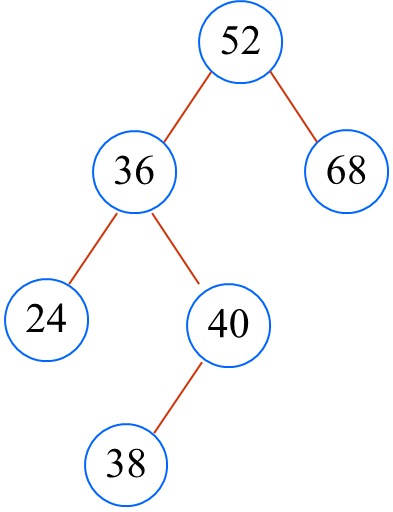


***Se compara el 38 con 52, como 38 es menor, vamos a la izquierda.***

***Se compara el 38 con 36, como 38 es mayor, vamos a la derecha.***

***Se compara el 38 con 40, como 38 es menor, vamos a la izquierda.***

***Como 40 no tiene sucesor izquierdo, ese es el lugar que le corresponde al 38.***



1. **BÚSQUEDA**

El algoritmo para buscar un dato en un Árbol Binario Ordenado, es el siguiente:

1. Repetir los pasos 2, 3 y 4.
2. Comparar el dato buscado con la raíz. Si son iguales, se termina la búsqueda.
3. Si el dato buscado es menor a la raíz, buscar en el sub árbol izquierdo.

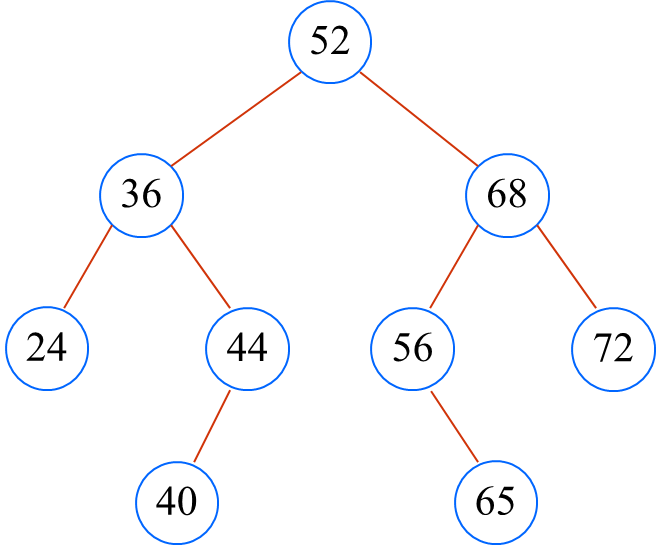
Si el nodo no tiene sucesor izquierdo, se termina la búsqueda.

1. Si el dato buscado es mayor a la raíz, buscar en el sub árbol derecho.

Si el nodo no tiene sucesor derecho, se termina la búsqueda.

***EJEMPLO:***

***En el siguient Árbol, busca el 56.***



***Se compara el 56 con 52, como 56 es mayor, buscamos en el sub árbol derecho.***

***Se compara el 56 con 68, como 56 es menor, buscamos en el sub árbol izquierdo.***

***Se compara el 56 con 56, como son iguales, se termina la búsqueda.***

1. **ELIMINACIÓN**

Consiste en sacar un nodo del Árbol. Luego de la eliminación, el Árbol debe continuar ordenado.

Dependiendo del número de sucesores del nodo a eliminar, el algoritmo varía.

En todos los casos, primero se busca el nodo a eliminar y se ubica al nodo padre.

* 1. Si el nodo a eliminar no tiene sucesores:

1. Si el nodo a eliminar es la raíz, se asigna null a la variable raíz y se termina el algoritmo.
2. Si el nodo a eliminar es el sucesor izquierdo del nodo padre, se asigna null al campo izq.
3. Si el nodo a eliminar es el sucesor derecho del nodo padre, se asigna null al campo der.
   1. Si el nodo a eliminar tiene un sucesor:
4. Si el nodo a eliminar es la raíz, el sucesor pasa a ser la raíz del Árbol.
5. Si el nodo a eliminar es el sucesor izquierdo del nodo padre, al campo izq se le asigna la dirección de memoria donde está ubicado el nodo hijo.
6. Si el nodo a eliminar es el sucesor derecho del nodo padre, al campo der se le asigna la dirección de memoria donde está ubicado el nodo hijo.
   1. Si el nodo a eliminar tiene dos sucesores:
7. Se ubica el predecesor in-orden.
8. Se intercambia el campo info del nodo a eliminar con el campo info del predecesor in-orden.
9. Se elimina el nodo del predecesor in-orden.

Este algoritmo también se puede aplicar si se toma el sucesor in-orden.

**CONCLUSIÓN**

Sólo en los Árboles Binarios Ordenados se pueden realizar operaciones adicionales al recorrido. En un Árbol Binario Ordenado, el recorrido in-orden muestra los elementos ordenados de menor a mayor. Luego de insertar o eliminar un nodo, el Árbol debe continuar ordenado.

**REFERENCIAS BIBLIOGRÁFICAS**

* Cairo, O.; Guardati, S. (2008). Estructuras de datos. 3ra. Edición. México D.F., México: McGraw Hill.
* Instituto NIIT (2011). Data Structures and Algorithms. Student guide.