

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

Campus Monterrey



Programación de estructura de datos y algoritmos fundamentales

Grupo 608

## **Reflex Act 2.3**

Profesor:

Eduardo López Benítez

Dan Herzberg Alperon


A00834217

Los BST's (Binary Search Trees) son considerados como uno de los fundamentales en la ciencia de la computación. Las propiedades que definen a los árboles binarios consisten en tener un nodo raíz, en donde se comienza el árbol, y que cada nodo tenga a lo más un hijo a la izquierda y uno a la derecha. El hijo de la izquierda siempre va a ser menor que el nodo padre y todo lo contrario con el hijo de la derecha, siempre va a ser mayor al padre. Estos tipos de estructuras de datos poseen características que son de gran utilidad para ciertos tipos de situaciones en donde se necesiten buscar datos de manera eficiente. Por la misma razón de que dentro de cada nodo del árbol existe un sub-árbol en donde todos los elementos a la izquierda son menores y a la derecha mayores. De esta manera las búsquedas son mucho más rápidas debido a que los datos ya se encuentran ordenados de una cierta manera.

Para la actividad utilizamos una solución que no incluía un BST, simplemente almacenamos los valores de las IP's en un vector, aplicamos un algoritmo de ordenamiento y desplegamos las IP's dentro del rango proporcionado por el usuario. Yo creo que el uso de los árboles binarios en esta solución hubiera sido una gran ventaja, en términos de eficiencia y rapidez. Principalmente porque esta estructura mantiene un orden, por lo que no es necesario recorrer todas las entradas para ordenarlas. Al momento de ingresar las IP's al árbol, automáticamente se estarían colocando en las posiciones correctas. Al igual se puede observar que el ordenamiento de dicha estructura no es de mucha dificultad, pero el acceder de manera secuencial a los datos es mucho más complicado que con las estructuras de datos lineales.

Una de las operaciones que es esencial para los BST's, es la de la incorporación de los datos a la estructura. Se tendría que hacer comparando el dato a agregar con los distintos nodos y que se haga el recorrido necesario hasta que se inserte como un nodo hoja. Su complejidad puede variar, si estamos hablando de un árbol que no está balanceado, esta operación se puede llevar a cabo de manera lineal, como en una Linkedlist, con complejidad de  $O(n)$ . En caso de tener un árbol balanceado, la complejidad sería de  $O(\log n)$ .

Referencias:

- Medina, R. (2020, 18 diciembre). *Estructura de Datos - Binary Tree (Arbol Binario)*. DEV Community . Recuperado 13 de octubre de 2022, de <https://dev.to/ronnymedina/estructura-de-datos-binary-tree-arbol-binario-2geb>