



**Instituto Tecnológico de Estudios Superiores de Monterrey,  
Campus Querétaro**

**TC1004B.501**

Programación de Estructura de Datos y Algoritmos Fundamentales

**Actividad**

Evidencia de BST

**Profesor**

Francisco Navarro

**Presenta**

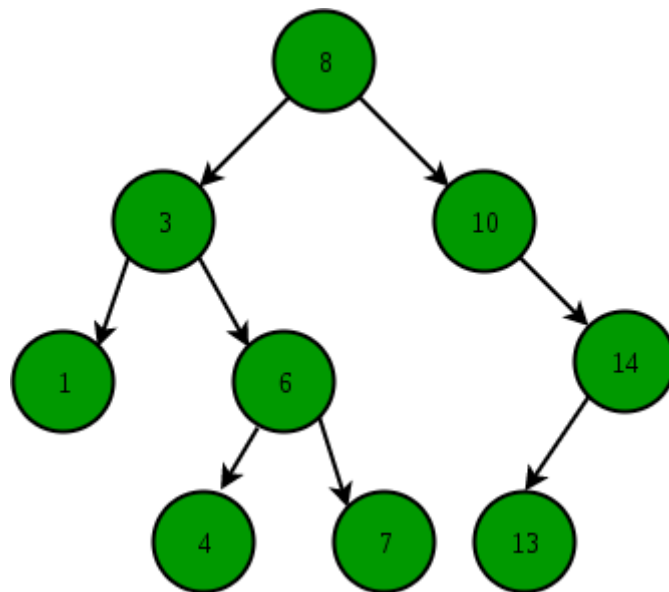
Carlos Rodrigo Salguero Alcantara

A00833341

Querétaro, Querétaro

Viernes 17, noviembre 2022

*Los Binary Search Tree son un tipo de binary tree que contiene las siguientes propiedades: el subárbol izquierdo de un nodo contiene sólo nodos con claves menores que la clave del nodo, el subarbol derecho de un nodo contiene sólo nodos con claves mayores que la clave del nodo y el subárbol izquierdo y derecho también debe ser un árbol de búsqueda binaria. (GeeksForGeeks, 2022)*



Los árboles binarios de búsqueda son utilizados en cualquier tipo de problema o situación que requiera de tener un orden jerárquico para ordenar los elementos.. De acuerdo con Binary Search Tree: Introduction, Operations, and Applications (s.f.), *“los BSTs son utilizados para la indexación y la indexación multinivel, para implementar varios algoritmos de búsqueda, para mantener un flujo ordenado de datos”*. De estos puedo dar ejemplo como dynamic sets, lookup tables y priority queues. Entre los algoritmos de ordenamiento se encuentra el tree sort.

Para situaciones como este problema, considero que son útiles para organizar de manera ordenada y por jerarquía los datos de los archivos de entrada y encontrar entre ellos el número de comparaciones que le tomó al compilador para organizar los datos.

Para concluir, considero que este tipo de estructura de datos son esencialmente útiles para ordenar de manera jerárquica y lógica los datos y operar sobre ellos con facilidad. Para este problema, yo utilicé un Binary Heap porque era una implementación más sencilla y eficiente que Binary Search Tree.

## Referencias

*GeeksforGeeks. (s. f.). Binary Search Tree.*

<https://www.geeksforgeeks.org/binary-search-tree-data-structure/>

*Binary Search Tree: Introduction, Operations and Applications. (s. f.).*

<https://afteracademy.com/blog/binary-search-tree-introduction-operations-and-applications>