Actividad 3.4 Alumno: Jesús Urquídez Calvo A00828368

25/10/2020 Docente: Luis Humberto González Guerra

Reflexión

Un BST (Binary Search Tree) es un conjunto finito de elementos dividido en subconjuntos. A estos se les conoce como subárboles y a cada elemento se le conoce como nodo. Un nodo puede tener hasta dos subárboles, uno derecho, uno izquierdo o en todo caso puede no tener ninguno. Estos son acomodados con los valores mayores del lado derecho del nodo padre y del lado derecho los menores. Gracias a esta estructura se puede manipular y acceder a los datos dentro de un árbol de manera más sencilla en cuanto a complejidad computacional y eficiencia.

Abordando la complejidad computacional, en la solución planteada se utiliza un add con una complejidad que depende de la altura del árbol. Esto nos permite manejar el problema desde una complejidad más sencilla que si se usara un vector o una lista. Gracias a la organización empleada en un BST el código se vuelve en general más rápido y menos repetitivo. Con esta idea en mente también es importante recalcar que no se implementa ningún algoritmo de ordenamiento como sería un quick sort, esto se hace directo desde el momento en el que se decide usar un BST pues se deben de seguir las reglas impuestas en cada nodo y sus hijos.

En general los árboles binarios de búsqueda son una excelente opción para organizar información que queremos tener ordenada de mayor a menor o viceversa. Esto es porque nos permite con mucha facilidad acomodar los datos con una prioridad, ya sea números mayores o números menores, cerca de la raíz de nuestro árbol. Este detalle ayuda de igual manera a la hora de querer mostrar los datos pues el print que se hace se concentra en los primeros datos del árbol (la raíz y aquellos nodos más cercanos a esta).