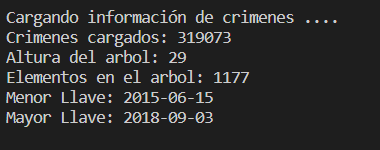
**OBSERVACIONES DEL LA PRACTICA**

Juan Sebastián Ortega Cod 202021703

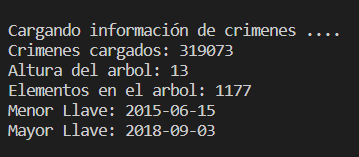
Yesid Camilo Almanza Cod 201921773

# **Preguntas de análisis**

1. ¿Qué diferencia existe entre las alturas de los dos árboles (BST y RBT)?



**Figura 1: Resultados carga del catalogo con BST**



**Figura 2: Resultados carga del catalogo con RBT**

Al realizar una comparación entre la Figura 1 y la Figura 2 es posible darse cuenta que bajo los mismos parámetros de búsqueda y con el mismo número de elementos elementos cargados existe una discrepancia notable entre la altura de ambos arboles. Mientras que el *Binary Search Tree* tiene una altura de 29, el *Red-Black Tree* tiene una altura menor a la mitad de la anterior, teniendo una altura de tan solo 13 para 1177 fechas.

1. ¿Por qué pasa esto?

Esto pasa debido a la forma en la que ambos tipos de arboles son construidos y a las reglas que se siguen en este proceso. En el caso del *Binary Search Tree* se parte de un nodo raíz (fecha del primer crimen en el catálogo) y se empiezan a comparar las fechas posteriores de forma que los elementos más grandes que el nodo anterior son ubicados a la derecha y los que tienen un valor menor son ubicados a la izquierda. De esta forma, es posible organizar el árbol de forma intuitiva puesto que para encontrar un elemento simplemente es necesario realizar comparaciones hasta llegar al elemento adecuado. Sin embargo, este método tiene un problema y es que su efectividad depende del orden de los datos que le son ingresados. Supongamos que se tiene una lista con N elementos organizados de menor a mayor, cuando esta lista sea convertida a un *Binary Search Tree* todos los elementos serán posicionados a la derecha del elemento anterior sin excepción, por lo que el elemento final se encontrará en una altura de N*.* Debido a esto, los *Binary Search Tree* no aseguran la creación de un árbol balanceado por lo que la organización de la información seguirá sin ser la óptima posible para este tipo de estructura de datos. Por otro lado, los *Red-Black Trees* cuentan con una serie de reglas especiales (además de la de añadir elementos mayores a la derecha y menores a la izquierda) que actualizan la estructura del árbol siempre que se añade un nuevo elemento. Estas reglas y el balanceo constante efectuado hacen que la altura final del árbol siempre esté aproximada a **log2 N** y en ningún momento llegará a los límites de **N.** En este caso particular es posible evidenciar esa diferencia puesto que **log2 1177 = 10.20** y la altura final de **13** es bastante cercana a este valor específico.