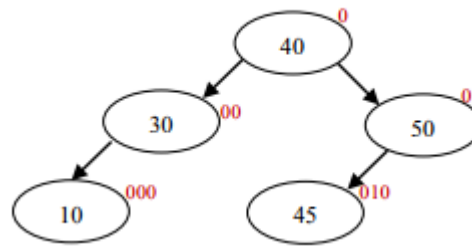


ACTIVIDAD 02: IMPLEMENTACIÓN DE OPERACIONES SOBRE ABBS



Juan Miguel Herrada Acosta

Realiza un seguimiento exhaustivo del método code (), tomando el árbol que se muestra en la siguiente figura como estructura de datos de ejemplo y partiendo de la frase [40, 10, 50].



```

private String code(BSTNode<T> current, T item, String codigo){
    if (current.nodeValue.compareTo (item)== 0) return codigo; 1

    if (current.nodeValue.compareTo (item)<0)
        return code (current.right,item,codigo+"1"); 2

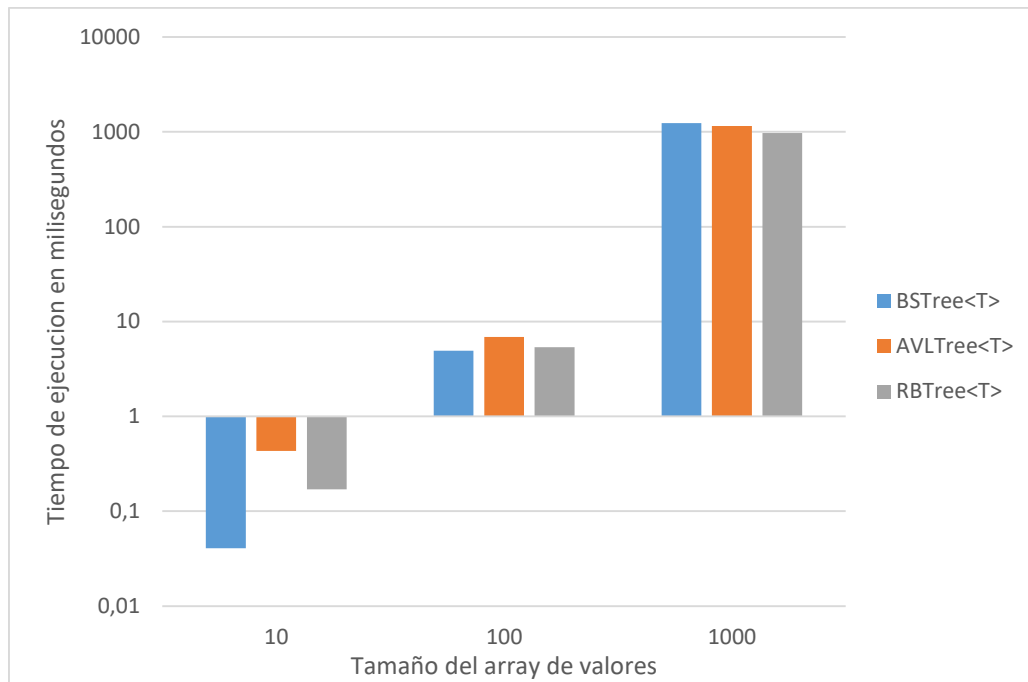
    return code (current.left,item,codigo+"0"); 3
}

```

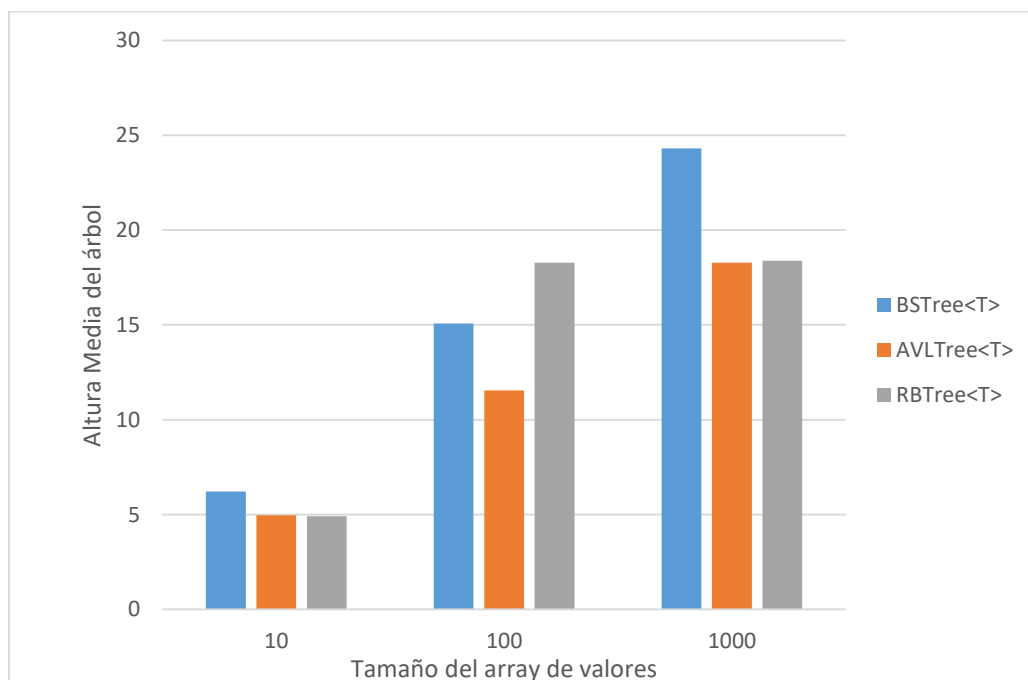
Current	Item	Código	Línea
40	40	0	1
40	10	0 0	3
30	10	0 00	3
10	10	0 000	1
40	50	0 000 0	2
50	50	0 000 01	1

- Vamos obtener y analizar distintos resultados modificando el tamaño de la muestra (10, 100, 1.000). Este proceso lo vamos a poder automatizar o, si preferís, vamos a hacer una ejecución distinta para cada tamaño de muestra. En la documentación tendréis que indicar, razonadamente, cuál ha sido vuestra decisión. A continuación, se comentarán los resultados obtenidos, basando vuestro análisis en una gráfica comparativa...de tiempos de ejecución y de altura media de cada estructura.

Realice una toma de datos para cada valor sin automatizarlas porque el hecho de automatizarlas no influiría en el tiempo de ejecución pero la implementación ya estaba realizada y aproveche esta misma.



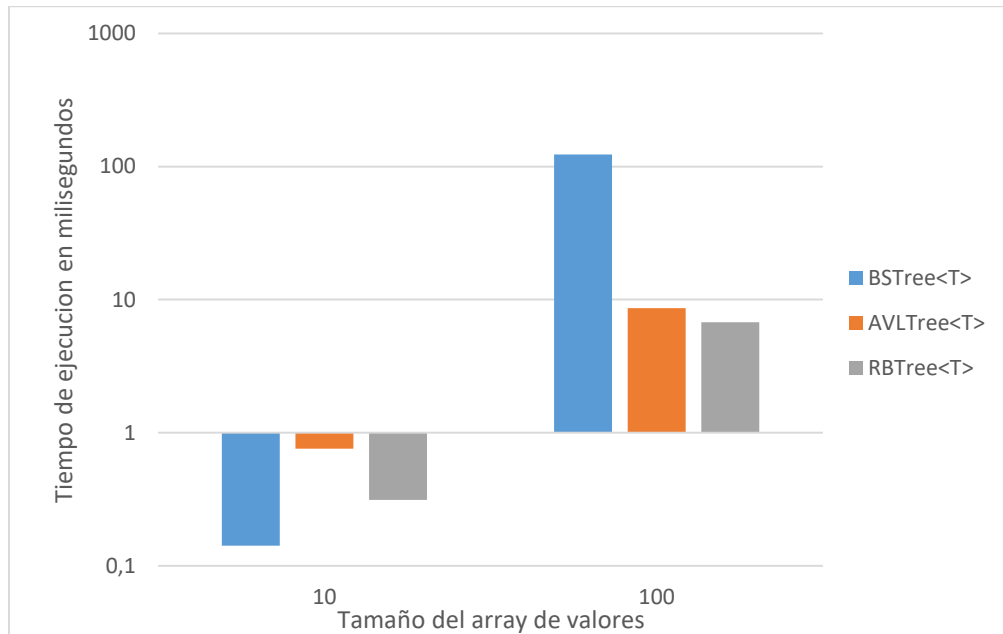
Como podemos observar en la gráfica al inicio con tamaños pequeños favorecen al BSTree pero cuando aumentamos el tamaño de entrada los valores se normalizan favoreciendo el liderazgo de RB con los mejores tiempos de ejecución.



Como podemos observar en la gráfica con tamaños bajos AVL y RB se comportan igual con tamaños medios sube un par de niveles RB sobre AVL y en tamaños más importantes BST se dispara y AVL con RB se mantienen en rangos similares.

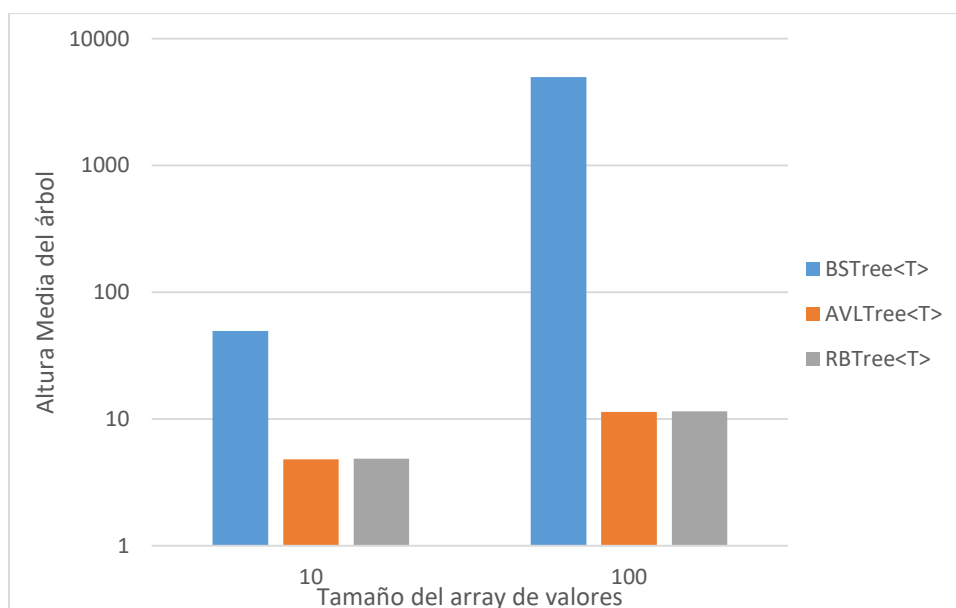
- - ¿Qué pasaría si eliminamos la instrucción `Collection.shuffle (datos)`? ¿Cuál sería el efecto en los resultados obtenidos? Comentad la funcionalidad de dicha instrucción y repetid el proceso anterior, comentando detalladamente cuáles han sido los nuevos resultados obtenidos y por qué. Basad vuestro análisis en gráficas comparativas.

Los valores tanto de tiempo como de altura y altura media de `BSTree` se disparan por su dependencia a que los datos estén ordenados y se degenera el árbol. En el caso de `AVL` y `RB` no habría ningún cambio significativo.



La grafica correspondiente a los tiempos de ejecución solo pusimos tamaño 10 y 100 ya que con 1000 consumía una cantidad de tiempo demasiado grande.

Como podemos observar el tiempo de `BST` se dispara cuando los valores están ordenados mientras que `AVL` y `RB` rondan valores similares.



La altura que adquiere BST se ve radicalmente superior a la altura que tienen tanto en el caso de AVL como RB, siendo siempre el desencadenante la aleatoriedad del valor de los datos insertados en el árbol.