



UNIVERSIDAD DE ALMERÍA

Estructuras de datos y Algoritmo I



Actividad 03

Curso: 2º Grado en Ingeniería Informática

Grupo docente: B1 **Grupo:** GTA3

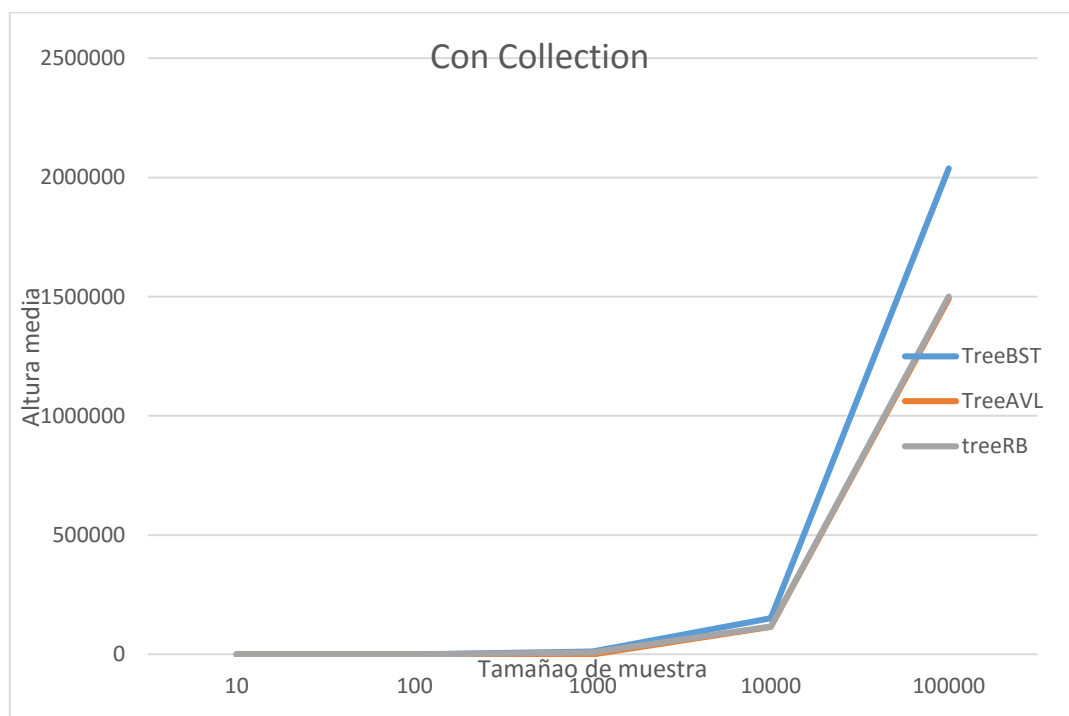
Nombre y Apellidos:

Jefferson Max Tomalá Villarreal

Ejercicio 7. Para comprobar el rendimiento de los árboles binarios de búsqueda vistos en clase (`BSTree<T>`, `AVLTree<T>` y `RBTree<T>`), se propone un test (`testPathHeightAllTress()`) que crea un `ArrayList<T>` con 100.000 números enteros diferentes y, a continuación, baraja dicha colección (`shuffle`). Una vez barajada la muestra, se insertan dichos elementos en cada estructura ABB. Después, procesamos el `ArrayList<T>` y, para cada elemento del mismo, llamamos a la función `pathHeight()` de su respectivo ABB, manteniendo un contador de su profundidad respecto a la raíz. Finalmente, se calcula la profundidad media (*average*) de un elemento en cada uno de los tres ABBs diferentes. Como veréis, este test **no es usual** (ya resuelto y con instrucciones tipo `syso()`) y lo único que hace es mostrar en Consola diferentes datos acerca de la altura de cada árbol, así como la media de altura de cada uno de los elementos que los conforman. Como ejercicio (**actividad03.pdf**) vamos a realizar lo que se indica a continuación:

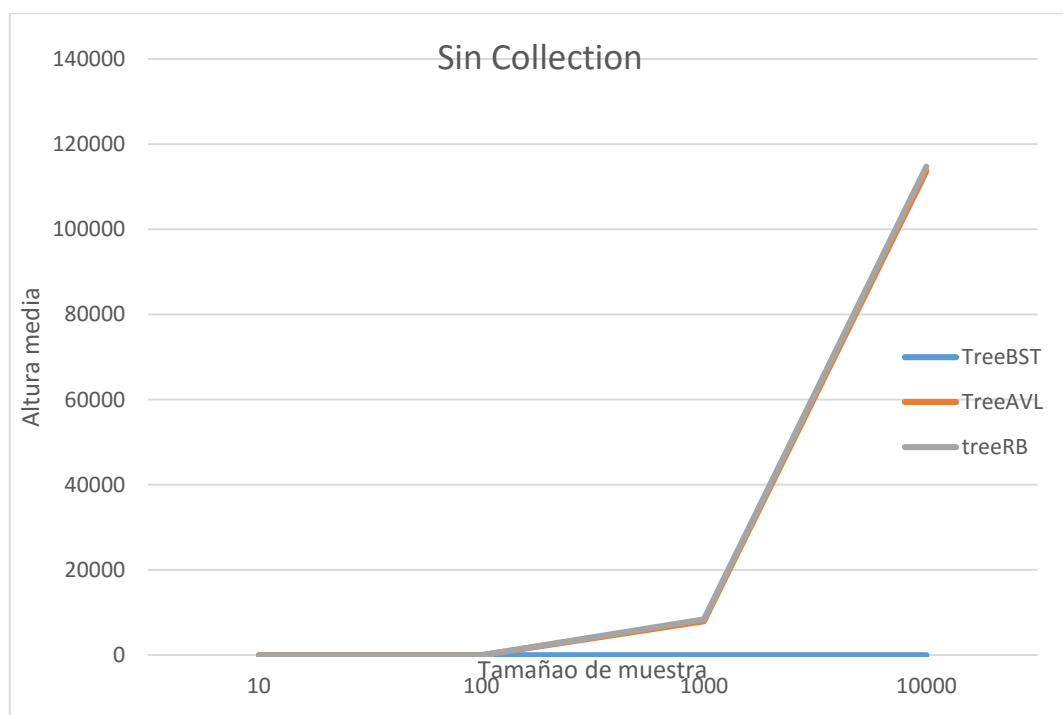
- Vamos obtener y analizar distintos resultados modificando el tamaño de la muestra (10, 100, 1.000, 10.000, 100.000). Este proceso lo vamos a poder automatizar o, si preferís, vamos a hacer una ejecución distinta para cada tamaño de muestra (modificando el valor de la variable *size*). En la documentación tendréis que indicar, razonadamente, cuál ha sido vuestra decisión. A continuación, se comentarán los resultados obtenidos, basando vuestro análisis en una gráfica comparativa.
- ¿Qué pasaría si eliminamos la instrucción `Collection.shuffle(datosE)`? ¿Cuál sería el efecto en los resultados obtenidos? Comentad la funcionalidad de dicha instrucción y repetir el proceso anterior, comentando detalladamente cuáles han sido los nuevos resultados obtenidos y por qué. Basad vuestro análisis en una gráfica comparativa.

Tamaño de la muestra	Con collection			Altura MEDIA		
	TreeBst	treeAVL	treeRB	TreeBst	treeAVL	treeRB
10	5	3	4	2.7	1.9	1.9
100	15	7	8	7.36	4.92	4.95
1000	24	11	13	12.104	8.2	8.257
10000	28	15	17	151.644	115.536	115.929
100000	40	19	21	2.037.482	1.490.913	1.501.076



En la gráfica se muestra que la altura media del BST es mayor. Esto es porque el AVL y RT te lo equilibra.

Tamaño de la muestra	SIN collection			Altura MEDIA		
	TreeBst	treeAVL	treeRB	TreeBst	treeAVL	treeRB
10	9	3	5	4.5	1.9	2.0
100	99	6	9	49.5	4.8	4.86
1000	999	9	16	499.5	7.987	8.381
10000	9999	13	20	4999.5	113.631	114.721
100000	Exception in thread "main" java.lang.StackOverflowError					



Sin el método collection los datos no te los ordena. Como vemos en la gráfica la altura media es de forma exponencial. En la gráfica la altura media del BST aumenta, debido a que se forma un árbol degenerado. Para el tamaño de la muestra de 100.000 no se ha podido realizar el main saliendo un mensaje de error: Exception in thread "main" java.lang.StackOverflowError.