

PRÁCTICA MARP

ÁRBOLES 2-3-4

- *Luis Pozas Palomo* -

1. Significado del formato de los ficheros de prueba

El objetivo de esta práctica es la implementación de los Árboles 2-3-4 con las operaciones fundamentales que son la búsqueda, la inserción y el borrado de elementos de tipo **entero**.

Los ficheros de prueba tendrán un **formato** como el siguiente:

Ejemplo 1:	Ejemplo 2:	Ejemplo 3:	Ejemplo 4:
<i>i</i> 7 <i>s</i> 7 <i>d</i> 7 <i>i</i> 6 <i>i</i> 7 <i>i</i> 8 <i>i</i> 9 <i>i</i> 10 <i>d</i> 7 <i>d</i> 6 <i>i</i> 5 <i>i</i> 3 <i>FIN</i>	<i>i</i> 6 <i>i</i> 5 <i>i</i> 4 <i>s</i> 4 <i>i</i> 3 <i>d</i> 3 <i>i</i> 1 <i>S</i> 1 <i>i</i> 7 <i>s</i> 7 <i>FIN</i>	<i>i</i> 8 <i>i</i> 9 <i>i</i> 10 <i>i</i> 11 <i>i</i> 12 <i>s</i> 12 <i>i</i> 2 <i>s</i> 2 <i>i</i> 20 <i>s</i> 20 <i>s</i> 11 <i>d</i> 11 <i>FIN</i>	<i>i</i> 7 <i>s</i> 7 <i>d</i> 7 <i>i</i> 4 <i>i</i> 3 <i>i</i> 2 <i>i</i> 1 <i>i</i> 10 <i>i</i> 11 <i>i</i> 9 <i>d</i> 4 <i>i</i> 1 <i>FIN</i>

Por cada par de líneas se realiza una operación de las mencionadas anteriormente, la “i” indica que es la **inserción**, la “s” indica la **búsqueda** y la letra “d” el **borrado** del elemento que aparece en la línea posterior de dichas operaciones, el caso de prueba acaba con la palabra “FIN”.

Para proceder a la **ejecución** de los ficheros de prueba y para facilitar el uso de los archivos creados, conviene modificar el Main cambiando el nombre de apertura de fichero al nuevo fichero de entrada, otra opción es ir copiando el contenido de los casos de prueba en el fichero llamado **casos.txt** que es el que va a abrir el programa por defecto.

2. Gráficas de tiempo de casos de prueba voluminosos.

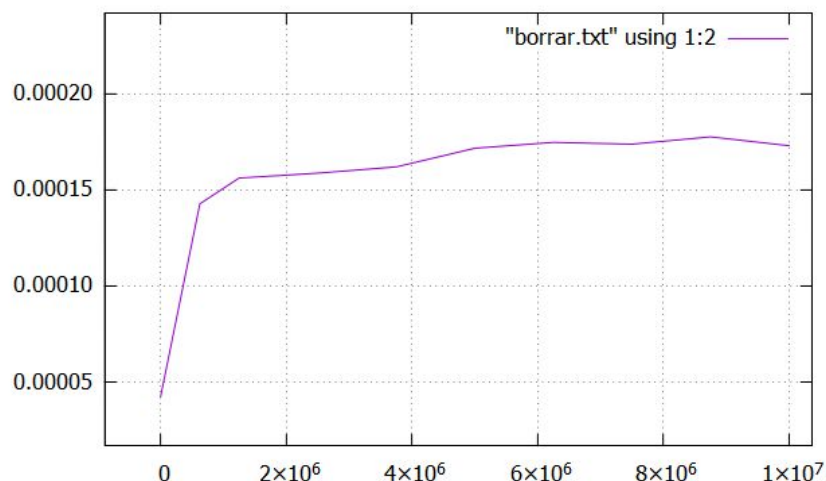
Tras la ejecución de 10 casos de prueba con árboles desde los 100 hasta 1.000.0000 elementos, consigo esta gráfica tras ejecutar 10.0000 veces (para conseguir valores más grandes que 10 milisegundos) la operación **buscar**, **borrar** e **insertar** y de esta manera poder realizar una media de lo que tardaría cada operación.

Puede observar en las gráficas que muestro a continuación que los costes de las operaciones son logarítmicos en el número de elementos de los árboles ya que las representaciones se asemejan a las funciones logarítmicas ya que crecen con valores pequeños y se estabilizan a un valor según van aumentando el tamaño de los árboles. Con lo que puede afirmar que el programa realizado cumple lo esperado por los costes teóricos de los árboles 2-3-4.

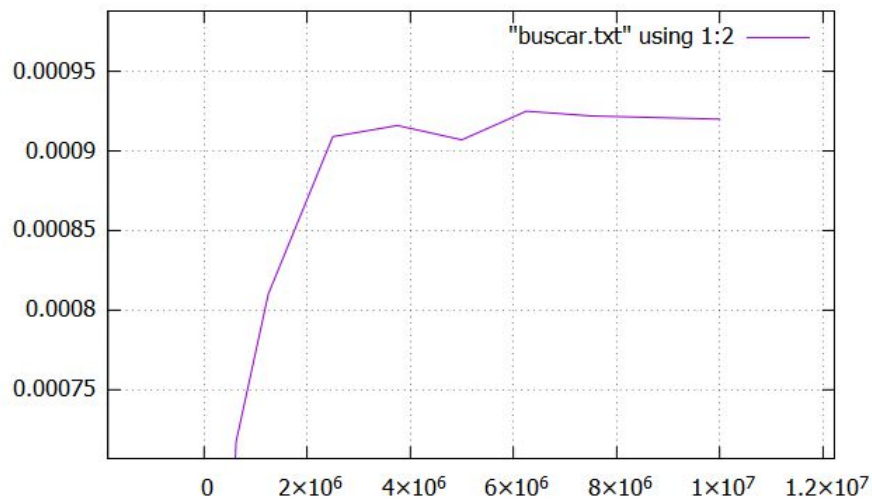
He de mencionar que las gráficas tienen pocos valores a representar puesto que los casos de prueba tienen numerosos elementos y he optado por coger 10 proporcionalmente para que de esta manera la gráfica quede representada en todo su dominio y así se perciba esa curva logarítmica mencionada anteriormente.

Para la generación de los casos de prueba representados en las gráficas, he realizado un código que insertaba los 'n' elementos de cada caso de prueba y a continuación la operación correspondiente repetida 1000 veces (para medir tiempos mayores de 10 milisegundos y realizar una media), **no** he creado ficheros de prueba tan voluminosos ya que me resultaba más cómodo trabajar con código.

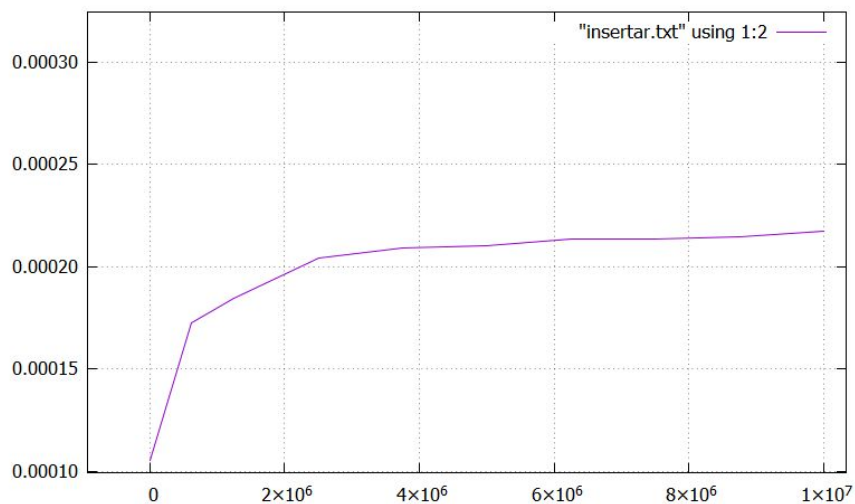
Gráfica correspondiente a la operación **borrar**:



Gráfica correspondiente a la operación **buscar**:



Gráfica correspondiente a la operación **insertar**:



En estas gráficas anteriores (que corresponden a las operaciones borrar, buscar e insertar respectivamente) los valores del eje de ordenadas están medidos en *milisegundos* y los del eje de abscisas representan los elementos que contiene cada caso de prueba.

A continuación voy a mostrar los valores representados en las gráficas de las tres operaciones implementadas que corresponden a los 10 casos de prueba voluminosos. El valor de la izquierda de las flechas representa el número de elementos que tiene el árbol (eje de abscisas) y el de la derecha el tiempo de ejecución de dicha operación en milisegundos (eje de ordenadas).

BUSCAR	BORRAR	INSERTAR
10000000 - 0.000920	10000000 - 0.0001730	10000000 - 0.0002171
8750000 - 0.000921	8750000 - 0.0001776	8750000 - 0.0002145
7500000 - 0.000922	7500000 - 0.0001738	7500000 - 0.0002138
6250000 - 0.000925	6250000 - 0.0001747	6250000 - 0.0002135
5000000 - 0.000907	5000000 - 0.0001717	5000000 - 0.0002100
3750000 - 0.000916	3750000 - 0.0001620	3750000 - 0.0002089
2500000 - 0.000909	2500000 - 0.0001587	2500000 - 0.0002045
1250000 - 0.000810	1250000 - 0.0001562	1250000 - 0.0001843
625000 - 0.000717	625000 - 0.0001428	625000 - 0.0001724
100 - 0.000216	100 - 0.0000424	100 - 0.0001056