Universidad Complutense de Madrid

Facultad de Informática

Métodos Algorítmicos en Resolución de Problemas I

Práctica Final

Árboles-B

Álvaro Corrochano López

Curso 2020/2021

Índice

1 Introducción

La práctica consiste en **representar Árboles-B** (siguiendo el libro "Cormen"[1]) con sus operaciones básicas: **inserción, búsqueda y eliminación.**

La representación entregada incluye algunas operaciones extra utilizadas para comprobar el corrector funcionamiento de los árboles, como el **recorrido** (función traverse) o si el árbol **está vacío**.

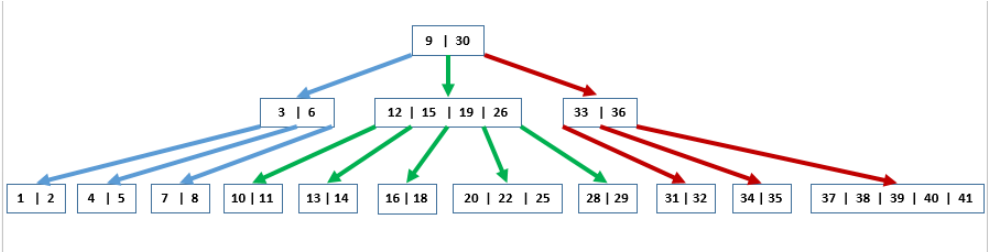


Figure 1: Ejemplo de Árbol-B

2 Pruebas

Para comprobar el funcionamiento, este se fue comprobando en **PruebaGeneral.cpp**, donde se iban probando los métodos según se implementaban. También se realizó otro archivo, **caseReader.cpp**, el cual lee un archivo de texto con el siguiente formato:

```
i
2
i
5
d
2
i
2
s
5
i
8
i
9
i
4
d
5
s
9
d
8
i
1
i
3
i
7
d
1
s
7
s
9
i
0
d
3
```

Figure 2: Muestra de un fichero de prueba

El programa lee el fichero (debe llamarse prueba.txt o bien cambiarse en el main del código) de forma que lee primero **la operación** y después **el número** a buscar, eliminar o insertar.

La **forma** en que interpreta las operaciones es:

- **i:** Insertar.
- **d:** Eliminar.
- **s:** Buscar.

3 Gráficas de Tiempos

Para las gráficas de ejemplo se utiliza un **tamaño de nodo = 3**, aunque este parámetro se puede **cambiar** en el código fácilmente **pasando un parámetro** al constructor, si no se le pasa nada, dará el tamaño por defecto de 3. Como mucho, el tamaño puede ser 1000.

Para medir los tiempos, se inserta, busca y elimina **50.000, 30.000 y 10.000** veces seguidas sobre un árbol con ese número de elementos dentro para poder conseguir valores de tiempo **por encima de los 10 ms** en las tres operaciones. En las gráficas observamos como se tarda menos de **45ms** en realizar **cualquiera de las operaciones** 50.000 veces. Las gráficas salen con forma de escalera debido a que con el aumento de 1 elemento el tiempo **apenas aumenta**, por ello tiene esta forma a pesar de tener 50.000 mediciones calculadas, porque la operación se realiza de forma **prácticamente inmediata**.

También se intentó medir cada una de las inserciones/búsquedas/eliminaciones **sin acumular el tiempo**, pero aún estando en valores de 50.000, **los valores eran muy bajos en general** (el ordenador los medía como 0) y era imposible determinar nada tal y como se observar en la figura de debajo, por lo que se decidió realizar únicamente estas gráficas (a lo largo del tiempo) en su lugar.

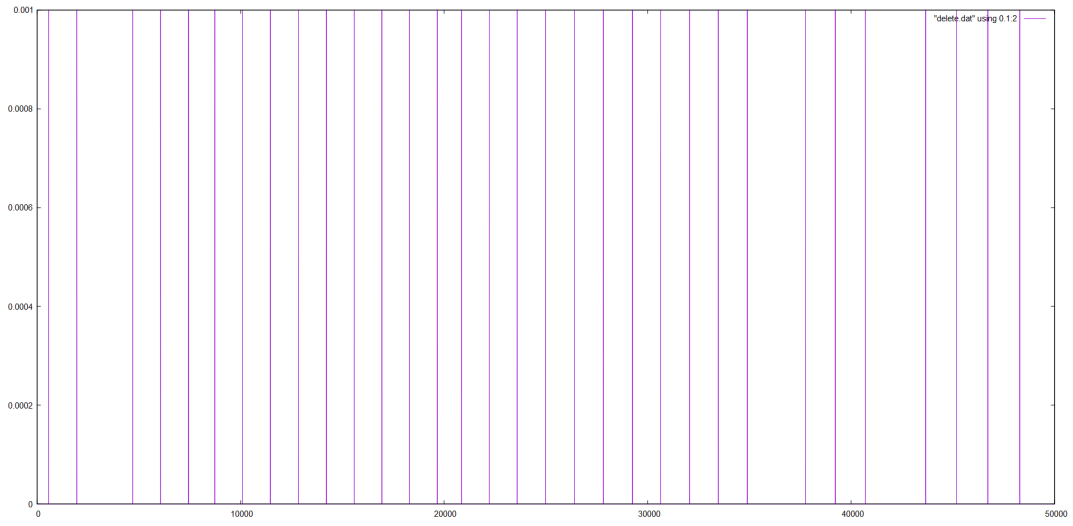
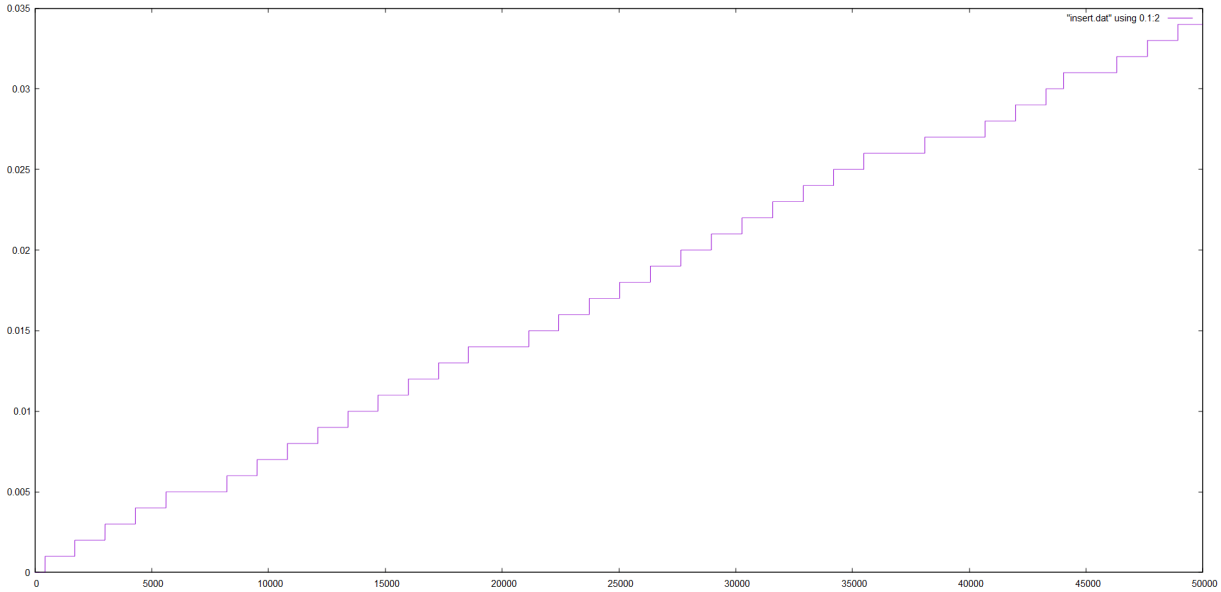
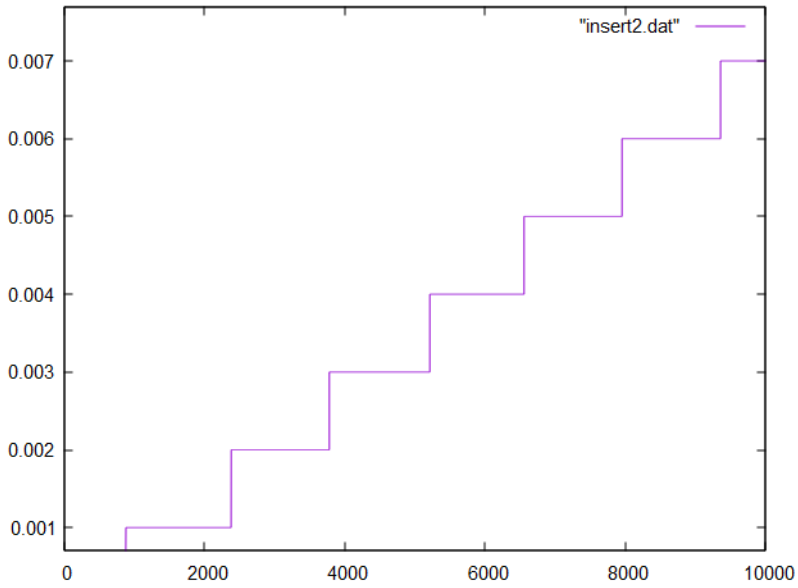
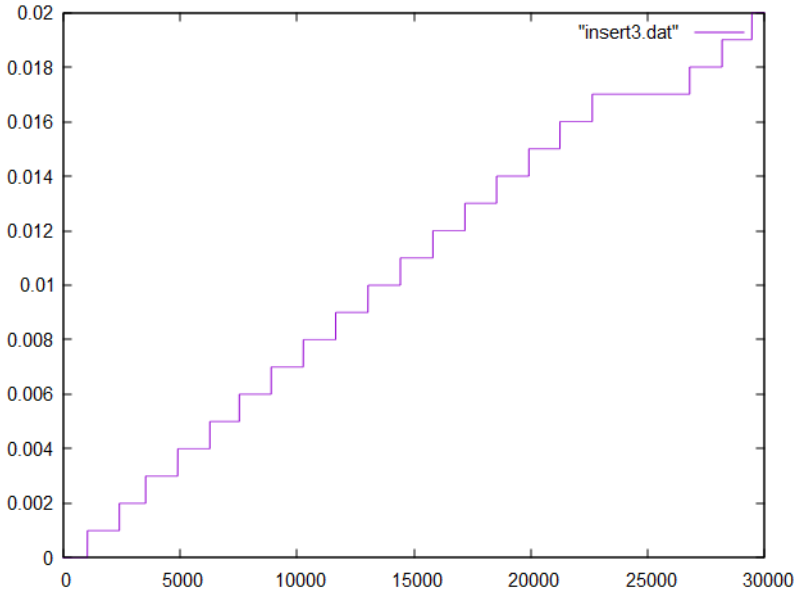


Figure 3: Ejemplo de medir el tiempo de cada vez que se aplica la operación por separado de 50.000 Deletes

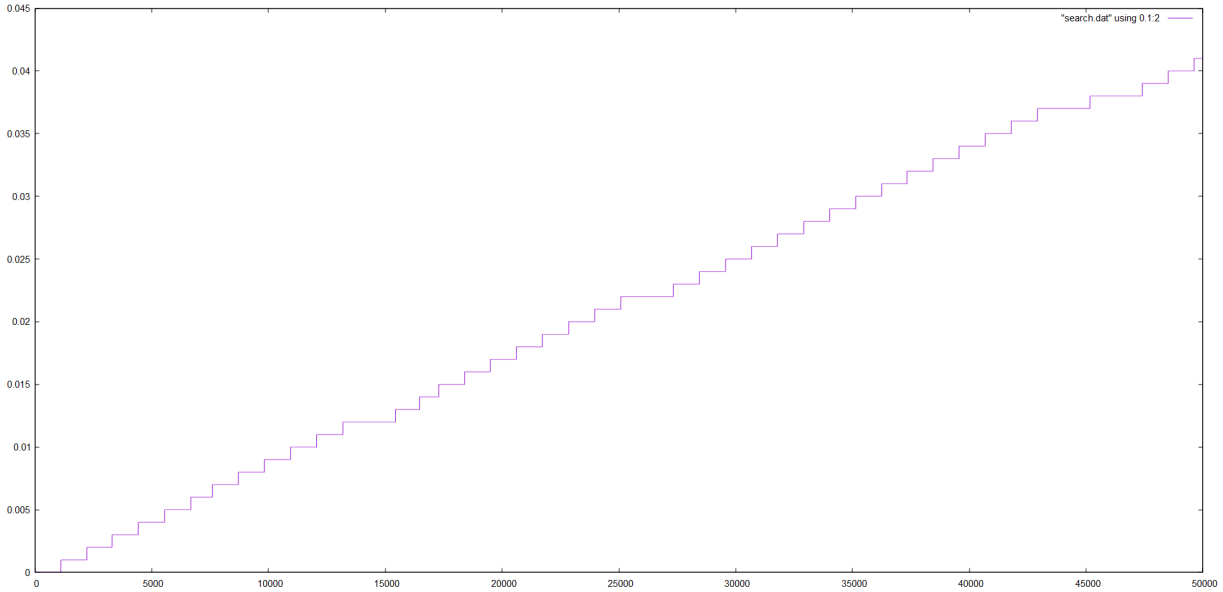
3.1 Gráficas de Insertar

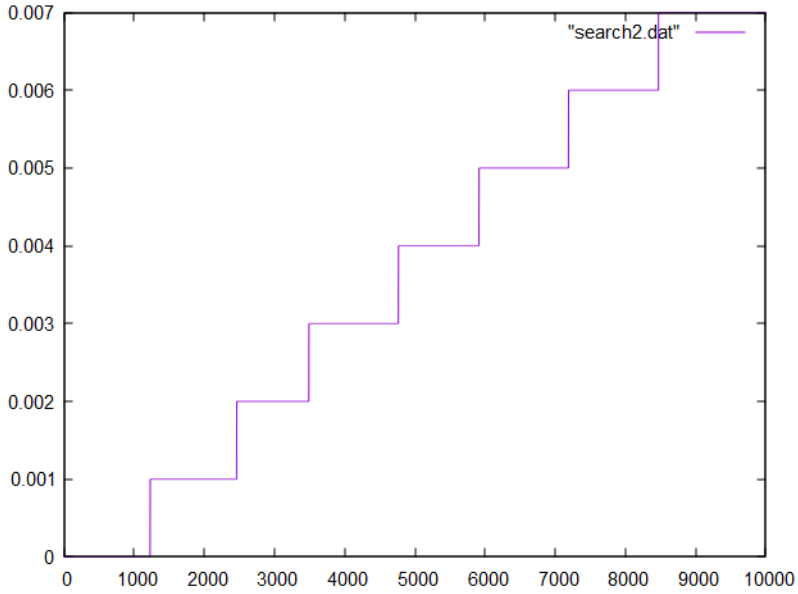


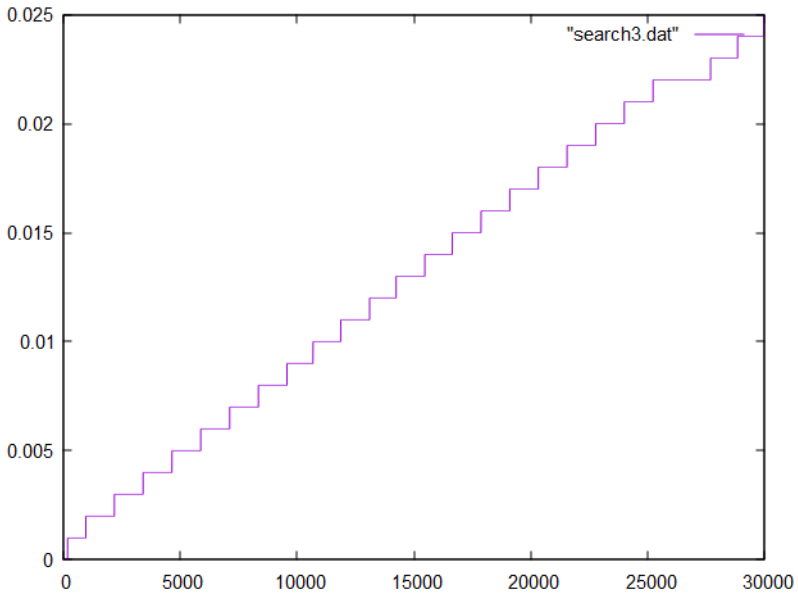



"insert3.dat"
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
5000
10000
15000
20000
25000
30000

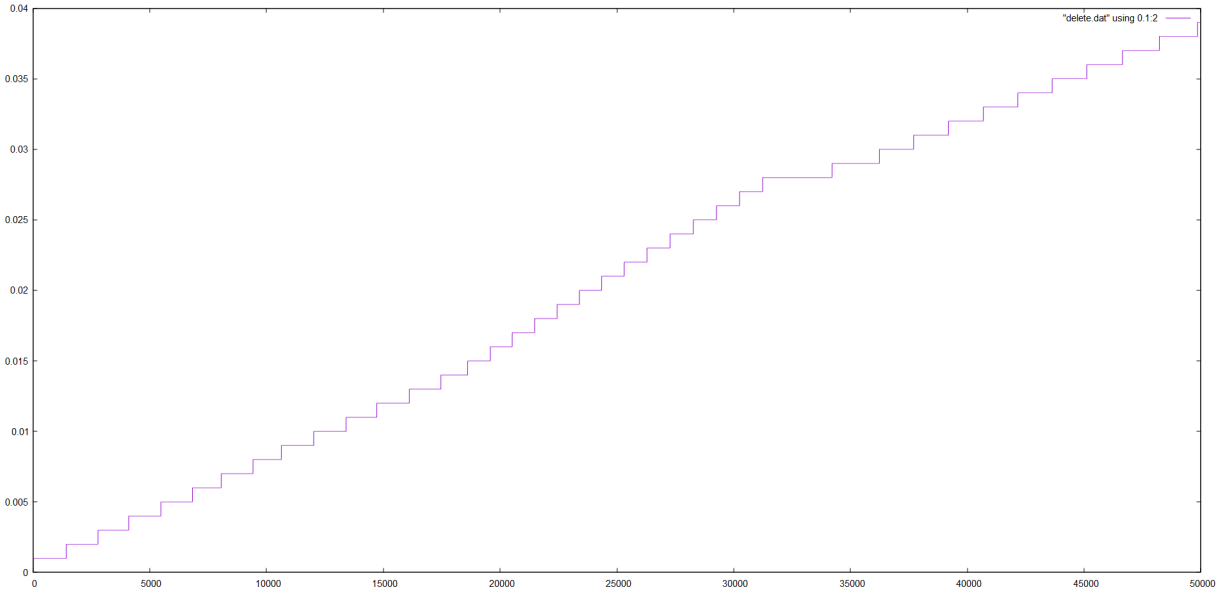
3.2 Gráficas de Buscar

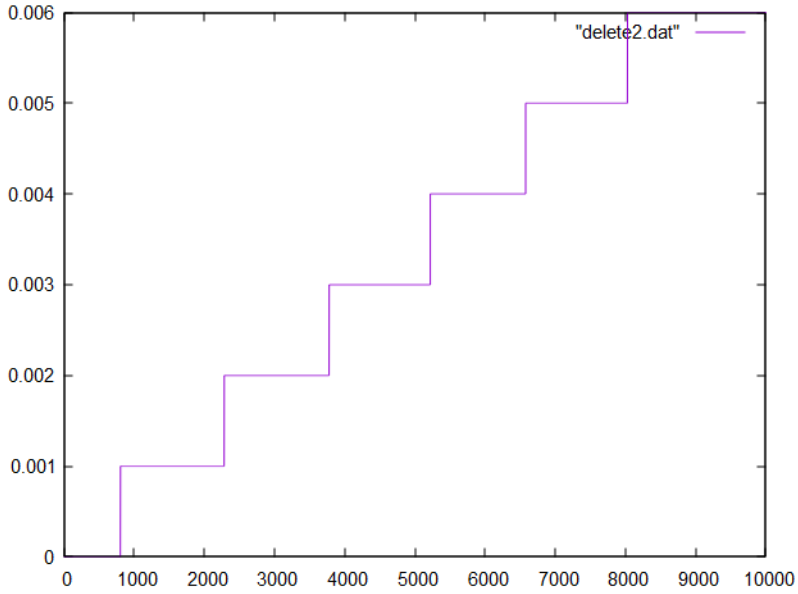


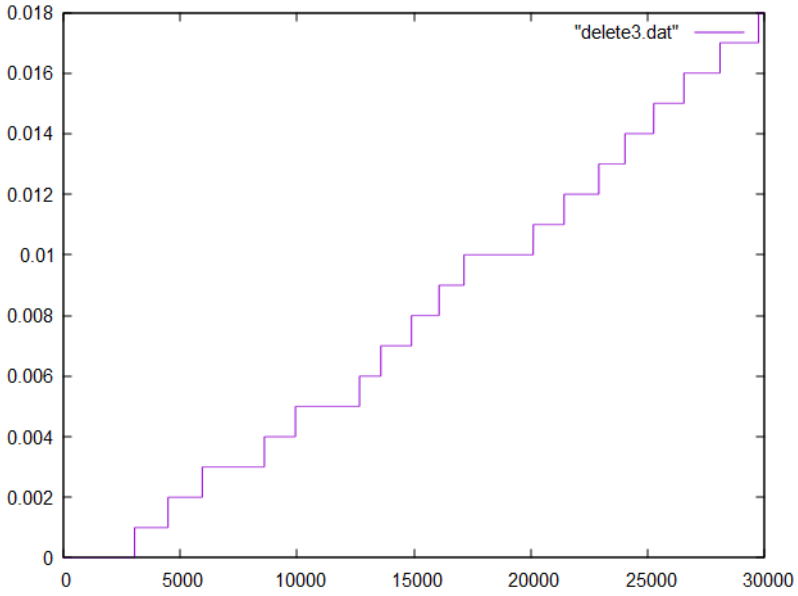



"search3.dat"
0.025
0.02
0.015
0.01
0.005
0
0
5000
10000
15000
20000
25000
30000

3.3 Gráficas de Eliminar






"delete3.dat"
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
5000
10000
15000
20000
25000
30000

3.4 Comparación de las Tres

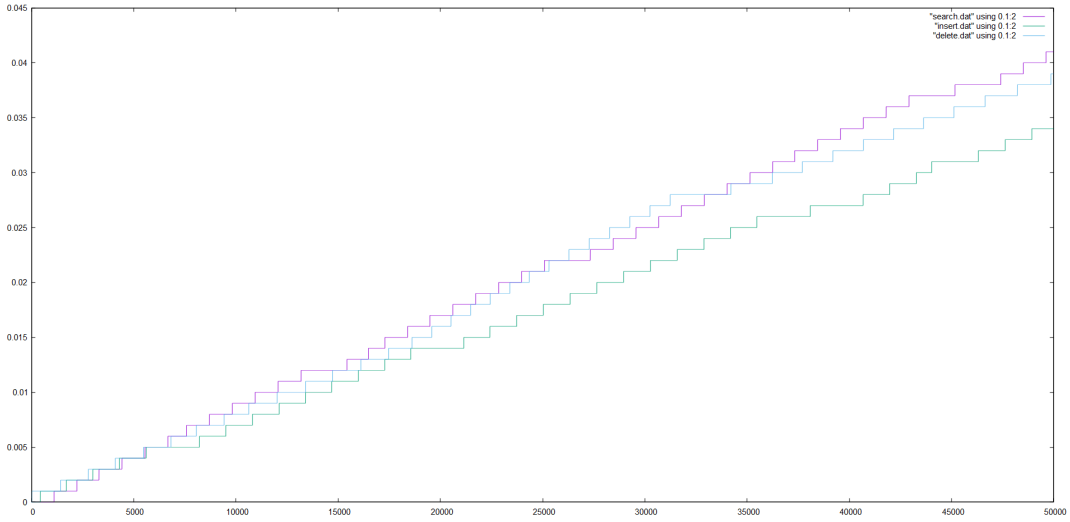


Figure 4: Gráfica de comparación de las tres ejecutando 50.000 veces cada operación

Se observa que la operación **más rápida** en lograr procesarse 50.000 veces es **insert** con una diferencia notable.
Destaca que la **más lenta** sea **search** a pesar de ser la que más rápido comienza.

Los costes estudiados en clase son de **O(log n) para las tres operaciones**, coste que observamos **cierto]**, pues aunque en la gráfica no se observa del todo bien, pues necesitaríamos realizar más veces cada operación (mi ordenador no permitía mucho más), el tiempo que se tarda en realizar cualquiera de ellas es **muy muy bajo**, como hemos observados en los ejemplos, por lo que concluyo que, efectivamente, **el coste de las tres operaciones es O(log n)**.

Referencias

[1] Thomas H. Cormen. Introduction to algorithms.