

Ejercicio 7: Multiplicación matricial

7.Implemente un programa que realice la multiplicación de dos matrices bidimensionales. Realice un análisis completo de la eficiencia tal y como ha hecho en ejercicios anteriores de este guión.

Para la realización de este ejercicio, hemos hecho un programa entero donde incluimos la clase matriz y un método que multiplica dos matrices.

El análisis teórico del algoritmo es el siguiente:

```
int Get(int valor1, int valor2){
    return datos[fils*valor1+valor2];
}

void Set(int valor1, int valor2, int tam){
    datos[fils*valor1+valor2]=random()%tam;
}

void multiplica(Matriz matriz2, int tam){
    int n = 0;
    Matriz resultado(tam,tam);

    for(int i=0; i<fils; i++){
        for(int j=0; j<fils; j++){
            for(int z=0; z<fils; z++){

                n+=matriz2.Get(i,z)*datos[z*fils+j];
            }
            resultado.Set2(i,j,n);
            n=0;
        }
    }
}

/*Antes de empezar con el cálculo de la eficiencia teórica,
cabe destacar, que todas las funciones que se llaman en el
algoritmo tienen una eficiencia O(1).
Dicho algoritmo está restringido a matrices n*n (fils*fils)por
lo que siempre van a tener el mismo número de filas y columnas(fils).
El primer for realiza n iteraciones y el segundo y el tercero
realizan también n iteraciones donde n varía según el script realizado.
El cuerpo del tercer for es O(1) por lo que por la regla del producto:
O(n*n*n*1)=O(n^3). Por lo que concluimos que tiene una eficiencia cúbica */
```

Primero compilamos el programa con el comando:

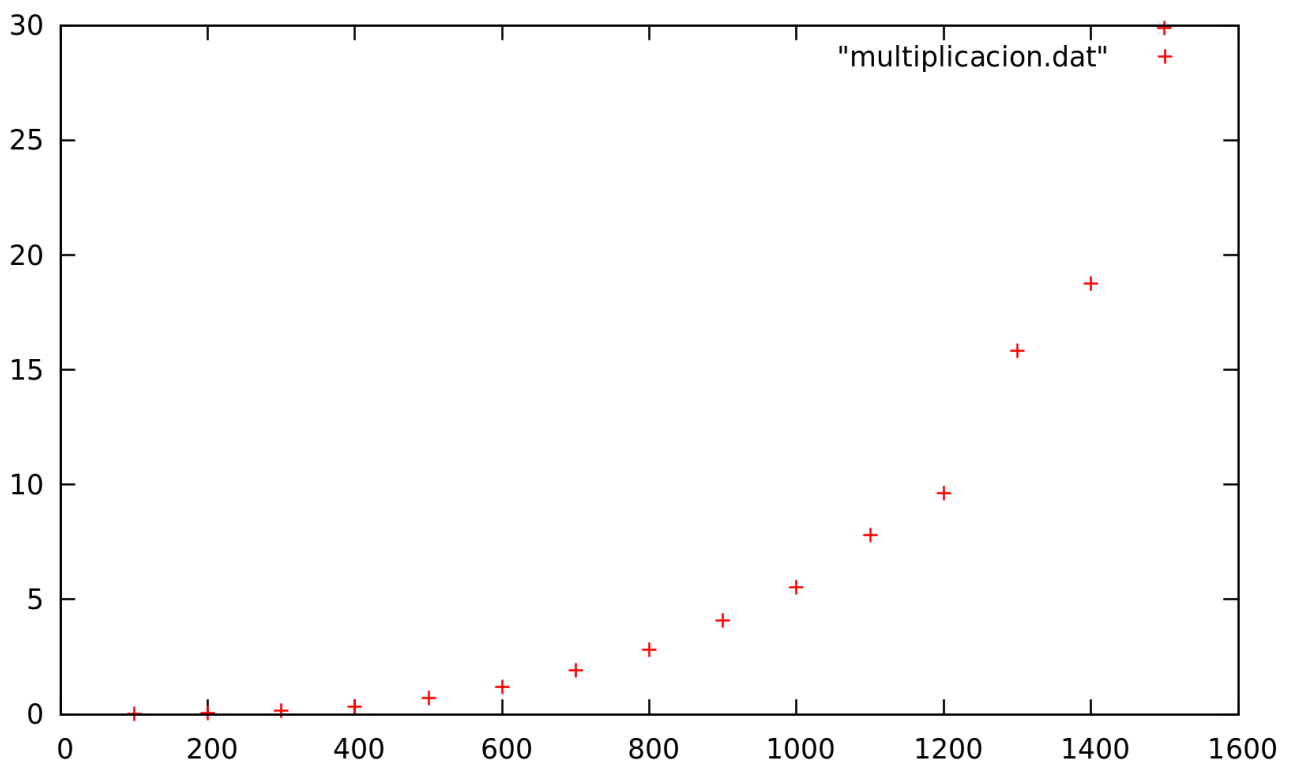
g++ multiplicacion.cpp -o multiplicacion

Para el cálculo de la eficiencia empírica, realizamos un script llamado 'multiplicacion.csh' que admite un parámetro que es la dimensión de la matriz, la cual varía de 100 a 1500.

Ejecutamos el script:

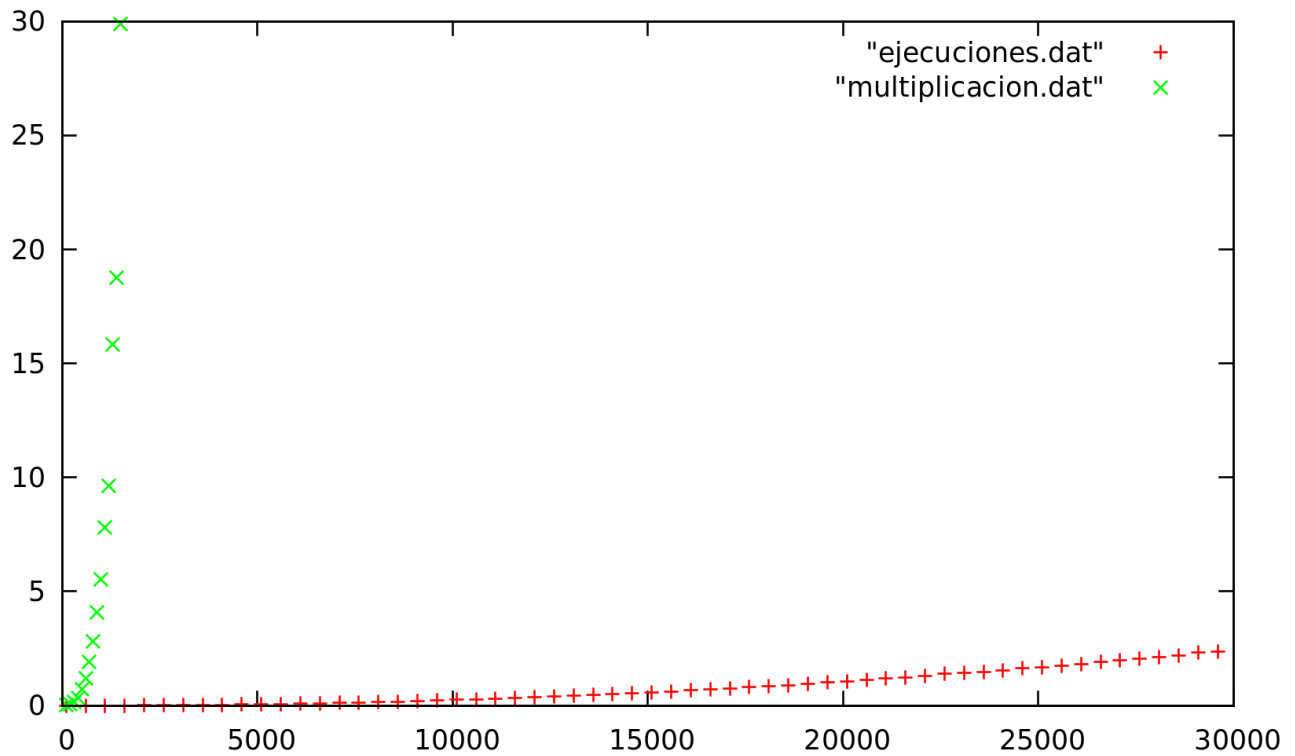
./multiplicacion.csh

Con la ayuda de gnuplot mostramos la eficiencia empírica, como ya realizamos anteriormente en otros ejercicios.



Como podemos observar, los datos obtenidos se corresponden con nuestras previsiones. Dicho algoritmo tiene una eficiencia de $O(n^3)$ y realiza una multiplicación matricial 1500×1500 en unos 30 segundos.

Comparemos ahora este algoritmo con el algoritmo del ejercicio 1:



Por tanto, podemos afirmar que la eficiencia del algoritmo del ejercicio 1 es mayor que la de este ejercicio ya que $O(n^2)$ es más eficiente que $O(n^3)$

Características del ordenador:

Fabricante	Lenovo
Procesador	Intel Core i7 4710HQ
Sistema Operativo	Ubuntu
Versión SO	14.04
RAM	8192MB
CPU	64 bits