

Práctica 1 ED, ejercicio 7

Javier Gálvez Obispo

Para la realización de la práctica se ha utilizado una máquina virtual con procesador i5-6500, con una frecuencia de reloj de 3.2GHz, limitado a una cpu, 3GB de RAM y ubuntu 16.04 LTS de 32bits como sistema operativo.

Código fuente:

```
#include <iostream>
#include <ctime>    // Recursos para medir tiempos
#include <cstdlib>  // Para generación de números pseudoaleatorios

using namespace std;

void sintaxis()
{
    cerr << "Sintaxis:" << endl;
    cerr << " TAM: Tamaño del vector (>0)" << endl;
    exit(EXIT_FAILURE);
}

int main(int argc, char * argv[])
{
    // Lectura de parámetros
    if (argc!=2)
        sintaxis();
    int tam=atoi(argv[1]);    // Tamaño del vector
    if (tam<=0)
        sintaxis();

    // Generación matrices
    int **v=new int*[tam];
    int **u=new int*[tam];
    int **w=new int*[tam];
    for(int i = 0; i < tam; i++){
        v[i] = new int [tam];
        u[i] = new int [tam];
        w[i] = new int [tam];
    }
    srand(time(0));           // Inicialización del generador de números pseudoaleatorios
    for (int i=0; i<tam; i++){
        for (int j = 0; j<tam; j++){
            v[i][j] = rand() % tam;
            u[i][j] = rand() % tam;
            w[i][j] = 0;
        }
    }

    clock_t tini;    // Anotamos el tiempo de inicio
```

```

tini=clock();

for(int i = 0; i<tam; i++){
    for(int j = 0; j<tam; j++){
        for(int k = 0; k<tam; k++){
            w[i][j]+=u[i][k]*v[k][j];
        }
    }
}

clock_t tfin; // Anotamos el tiempo de finalización
tfin=clock();

// Mostramos resultados
cout << tam << "\t" << (tfin-tini)/(double)CLOCKS_PER_SEC << endl;

    for (int i = 0; i<tam; i++){
        delete [] u[i];
        delete [] v[i];
        delete [] w[i];
    }
    delete [] w;
    delete [] u;
    delete [] v;
}

```

Para la compilación del programa se ha utilizado la siguiente orden:
g++ matrices.cpp -o matrices

El script utilizado para las ejecuciones es el siguiente:

```

#!/bin/bash
inicio=100
fin=1000
incremento=20
ejecutable="matrices"
salida="matrices.dat"

i=$inicio
echo > $salida
while [ $i -lt $fin ]
do
    echo "Ejecución tam = " $i
    echo `./$ejecutable $i` >> $salida
    i=$((i+$incremento))
done

```

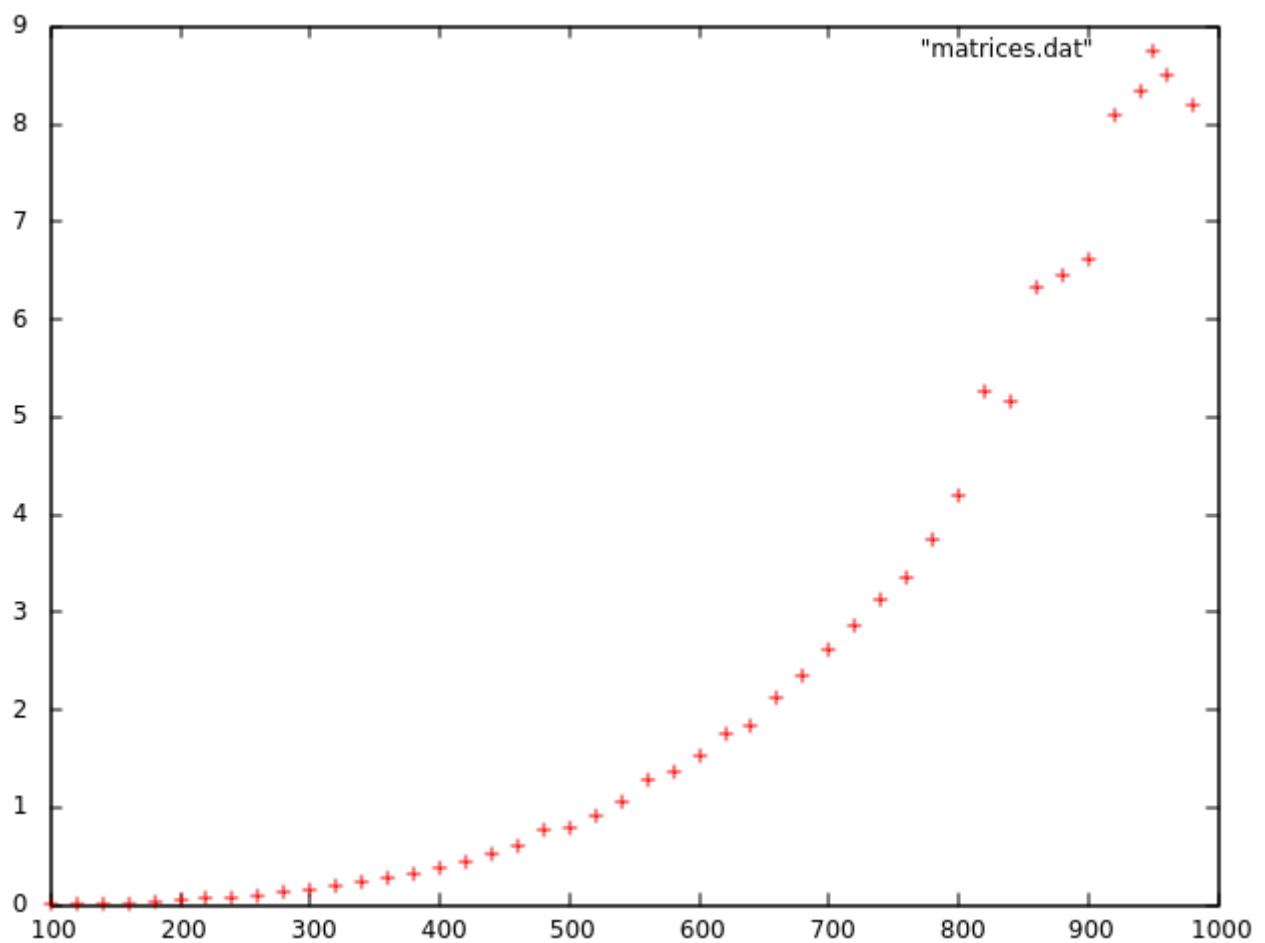
Eficiencia teórica:

```
for(int i = 0; i<tam; i++){  
    for(int j = 0; j<tam; j++){  
        for(int k = 0; k<tam; k++){  
            w[i][j]+=u[i][k]*v[k][j];  
        }  
    }  
}
```

				$O(n^3)$
		$O(n)$	$O(n^2)$	
$O(1)$				

$$O(n) * O(n) * O(n) = O(n^3)$$

La gráfica al dibujar los datos obtenidos tras la ejecución es:



Hallamos la curva que mejor se acerca de la forma Ax^3+Bx^2+Cx+D

a = 1.71342e-08
b = -9.42493e-06
c = 0.00229935
d= -0.155698

Y obtenemos la siguiente gráfica:

