

Práctica 1: Eficiencia

Antonio Manuel Fernández Cantos

12 de octubre de 2015

Índice

1. Introducción	1
2. Componentes utilizados	1
3. Ejercicio 7: Multiplicación matricial	2
3.1. Introducción	2
3.2. Código	2
3.3. Eficiencia teórica	3
3.4. Eficiencia empírica	4
3.5. Ajuste por regresión	4

1. Introducción

La práctica consiste en calcular la eficiencia **teórica y empírica** de un código en c++ y **realizar un ajuste de la curva de eficiencia teórica a la empírica**. Se utilizará la biblioteca **ctime** para poder obtener los resultados empíricos. Dentro de la biblioteca ctime tenemos la función **clock()** que devuelve el número de ticks que han transcurrido desde un momento determinado, es esta función la que usaremos para medir la diferencia de tiempo entre el inicio del algoritmo y su finalización.

2. Componentes utilizados

En el cálculo empírico, el algoritmo tardará más o menos en función de:

- **Hardware usado:**

- CPU
- RAM
- HDD

- **Sistema Operativo**
- **Compilador (y sus opciones de compilación)**
- **Bibliotecas**

Todos estos componentes se tienen en cuenta cuando se obtiene el tiempo que tarda nuestro algoritmo en ejecutar todas las sentencias. Dependiendo de la potencia de nuestro ordenador y de las librerías usadas, el algoritmo tardará más o menos. Para la realización del cálculo empírico de los ejercicios, he usado los siguientes componentes:

- **Hardware usado:**
 - Procesador: 8x Intel(R) Core(TM) i7-3630QM CPU@2.40MHz
 - RAM: 6GB
 - CPU clock: 1200 MHz
 - HDD: 750GB
- **Sistema Operativo**: Ubuntu 14.04.3 LTS
- **Compilador**: GCC sin opciones de compilación
- **Bibliotecas**:
 - iostream (E/S)
 - ctime (Para medir el tiempo de ejecución de un algoritmo)
 - cstdlib (Para generar números pseudoaleatorios)

3. Ejercicio 7: Multiplicación matricial

3.1. Introducción

El ejercicio consiste en realizar la multiplicación de dos matrices bidimensionales (dos dimensiones) y medir su eficiencia teórica y empírica. Para entender como se hace la multiplicación de matrices, es recomendable visitar el siguiente enlace: https://es.wikipedia.org/wiki/Multiplicaci%C3%B3n_de_matrices

3.2. Código

Las partes del código que he usado y que resultan interesantes comentar son las siguientes:

```

1 void Multiplicar(int **nueva, int **m1, int **m2, int tam){
2
3     for(int i=0;i<tam;i++){
4         for(int j=0;j<tam;j++){
5             nueva[i][j]=0;
6             for(int k=0;k<tam;k++){
7                 nueva[i][j]+=m1[i][k]*m2[k][j];
8             }
9         }
10    }

```

Como podemos observar, la función recibe tres punteros a punteros a enteros (tres matrices) y el tamaño de la matriz, las matrices son **cuadráticas**. Sobre la multiplicación, a la posición (i,j) se le asigna la fila iesima de la primera matriz por la columna jesima de la segunda matriz.

El siguiente código muestra como se han declarado las matrices y la asignación de datos a las matrices.

```

1 int **m1, **m2, **nueva;
2 m1 = new int*[tam];
3 m2 = new int*[tam];
4 nueva = new int*[tam];
5
6 for(int i=0; i<tam; i++){
7     m1[i]=new int[tam];
8     m2[i]=new int[tam];
9     nueva[i]=new int[tam];
10 }
11
12 srand(time(0));
13 for(int i=0; i<tam; i++){
14     for(int j=0;j<tam;j++){
15         m1[i][j] = rand() % tam;
16         m2[i][j] = rand() % tam;
17         nueva[i][j]=0;
18     }
19 }

```

Por último vamos a mostrar donde se calcula los tiempos a medir:

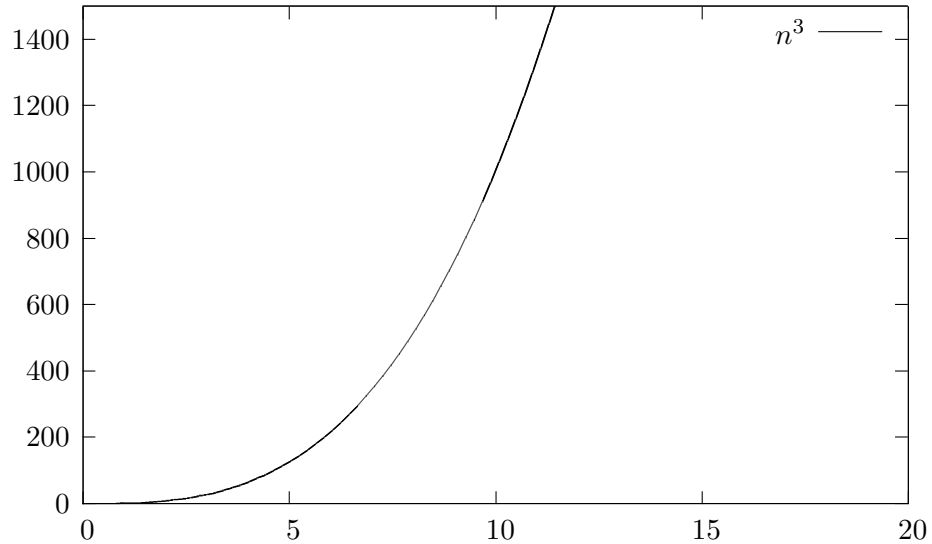
```

1     clock_t tini;    // Anotamos el tiempo de inicio
2     tini=clock();
3
4     Multiplicar(nueva,m1,m2,tam); // Llamamos a la
        multiplicacion de matrices
5
6     clock_t tfin;    // Anotamos el tiempo de finalizaci n
7     tfin=clock();

```

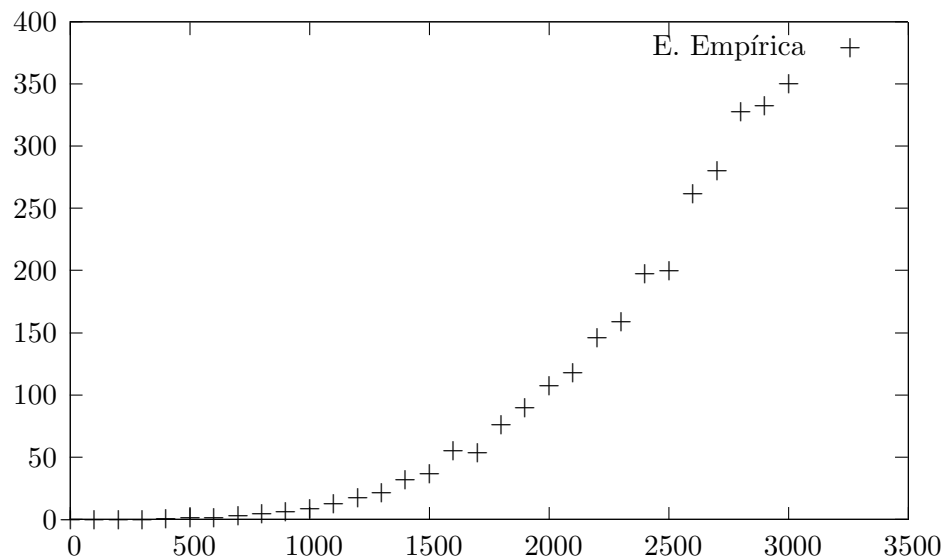
3.3. Eficiencia teórica

El trozo de código que nos interesa analizar es la función void Multiplicar(...). Como bien sabemos cada bucle for es $O(n)$. Al tener tres bucles for anidados y usando la regla del producto obtenemos que la eficiencia teórica de la función es $O(n^3)$. La gráfica de $O(n^3)$ es la siguiente:



3.4. Eficiencia empírica

Para medir la eficiencia empírica he comenzado con matrices 2x2, se han ido aumentando 100x100 hasta alcanzar los 3002x3002. La gráfica que he obtenido ha sido la siguiente:



3.5. Ajuste por regresión

En el ajuste por regresión, considero que $f(x)$ es de la forma $a * x^3 + b * x^2 + c * x + d$. Los valores de a , b , c y d son los siguientes:

$a = 1,1764 * e^{-08}$ $b = 1,25356 * e^{-05}$ $c = -0.0207087$ $d = 4.05516$

Y su gráfica es la siguiente:

