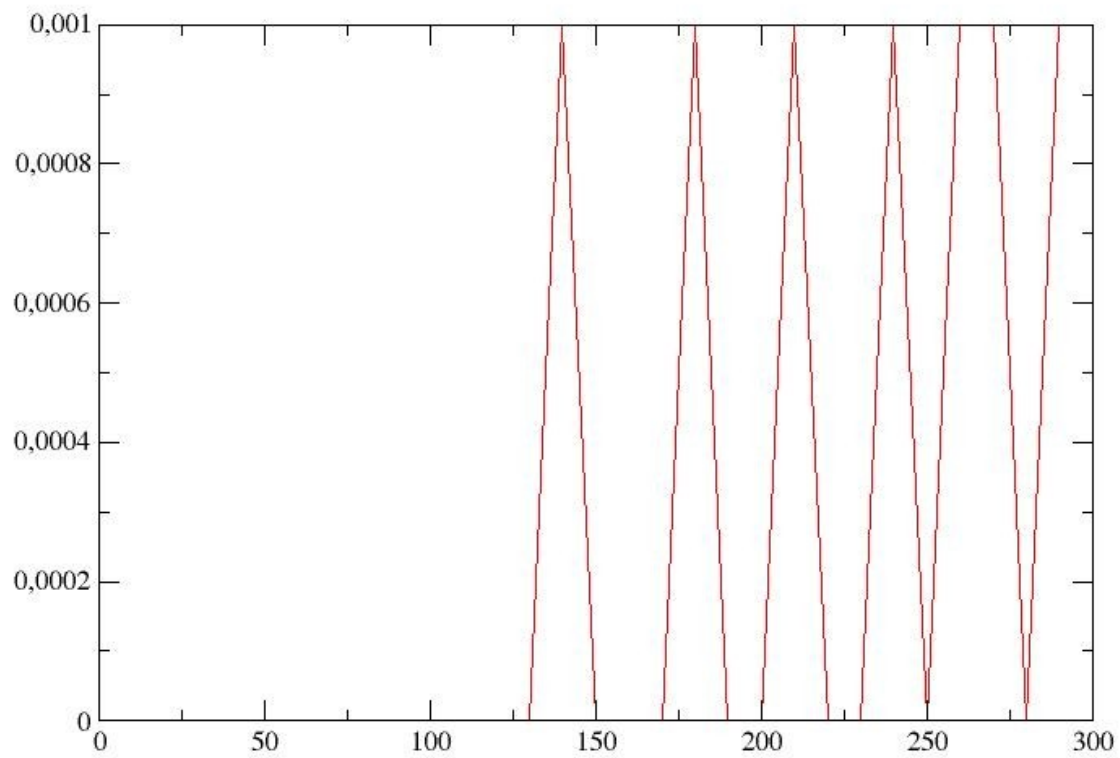


PRÁCTICA 1

Cálculo de la eficiencia

Álvaro Maximino Linares Herrera



Problema 1:

El primer problema que presento es uno muy sencillo, se trata de rellenar matrices cuadradas con números aleatorios.

Código:

```
void RellenaAuto(int **m, int tam){  
  
    for(int i=0; i<tam; i++)  
        for(int j=0; j<tam; j++)  
            m[i][j]=rand()%20; //Inicia Matriz entre 0 y 20 aleatoriamente  
  
}
```

Este sería el código de para rellenar automáticamente una matriz de tamaño nxn. Como se puede apreciar es un código bastante simple, basta con dos bucles for y una asignación.

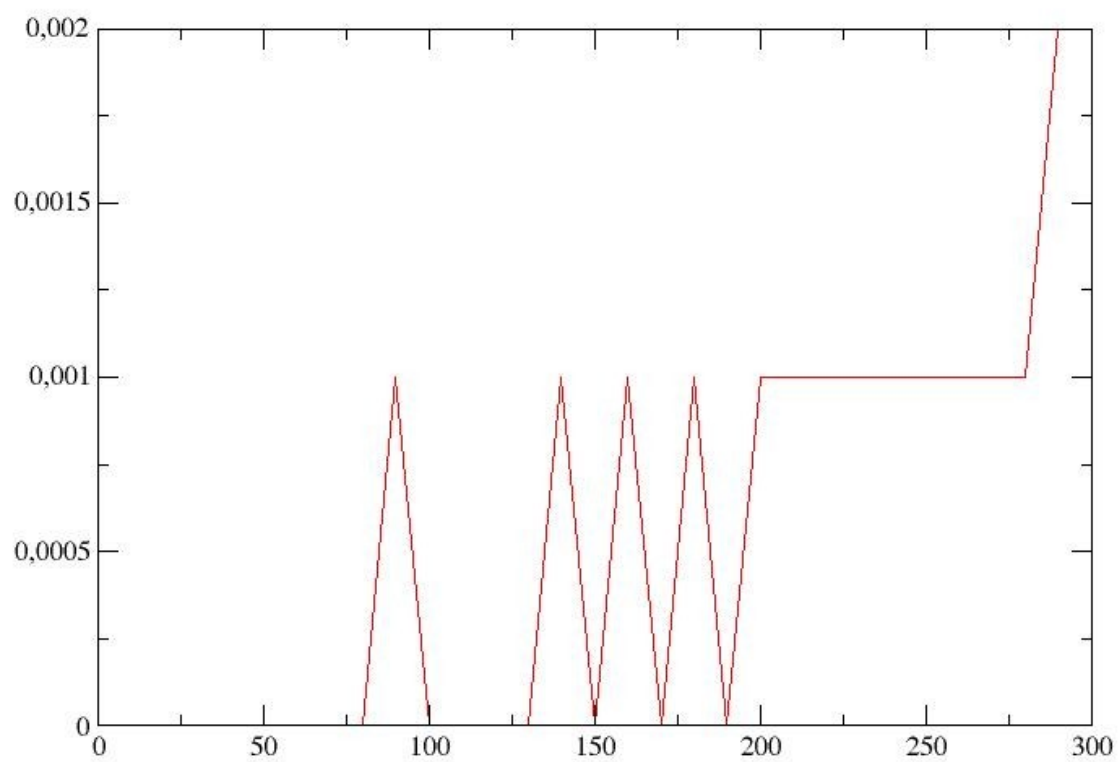
Cálculo de la eficiencia teórica:

Bien, como vemos son dos bucles for, con dos accesos (a filas y columnas), una asignación y un modulo, esto nos daría 4 instrucciones, esta porción de código la acotaremos con una constante a.

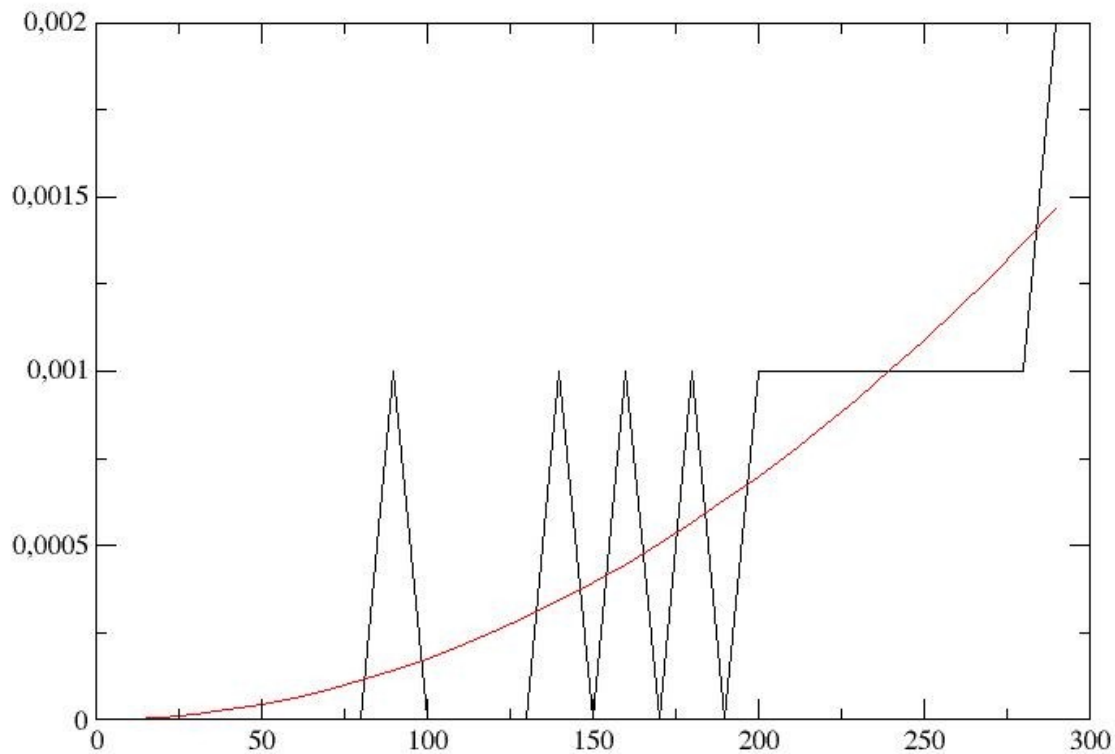
$$\sum_{i=1}^n \sum_{j=1}^n (4) = 4 \sum_{i=1}^n n = 4n^2 \in O(n^2)$$

Cálculo de la eficiencia empírica:

Como indicación he modificado la macro para que en lugar de que se ejecute el número de veces que usted puso se ejecute solo 300 veces.



Cálculo de la eficiencia híbrida:



Y este sería el cálculo de la eficiencia híbrida.

Para ello en el programa le metí como usted indicó en la práctica 0 el número de constantes ocultas, en mi caso solo una, a_0 , de modo que $y = a_0 * x^2$.

Problema 2:

El segundo problema consiste en la suma de dos matrices, es otro ejercicio simple que se hace mediante el código:

Código:

```
void **SumaMatrices(int **x, int **y, int **&res, int tam){
    for(int i=0; i<tam; i++)
        for(int j=0; j<tam; j++)
            res[i][j]=x[i][j]+y[i][j];
}
```

Es parecido al anterior en cuanto a complejidad de realización.

Trata que a partir de tres matrices y el tamaño mediante dos bucles for calculamos la suma de las dos primeras matrices cuadradas en la tercera matriz.

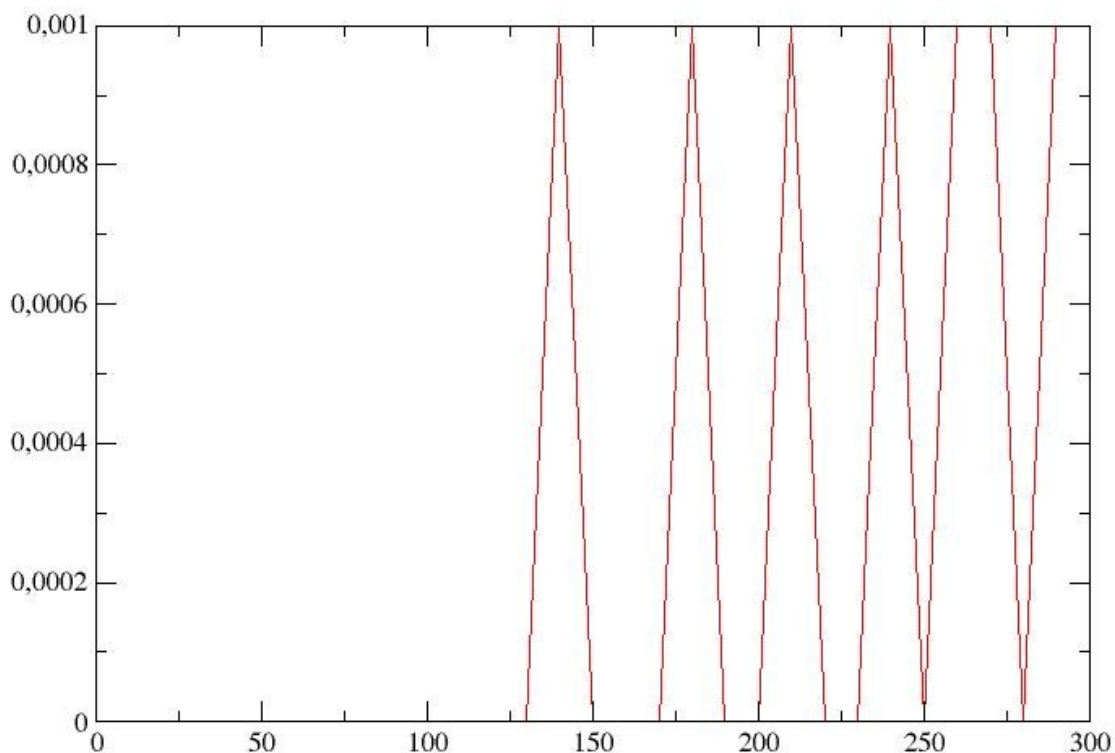
Cálculo de la eficiencia teórica:

Bien, como vemos son dos bucles for, con dos accesos (a filas y columnas), una asignación y un modulo, esto nos daría 4 instrucciones, esta porción de código la acotaremos con una constante a.

$$\sum_{i=1}^n \sum_{j=1}^n (8) = 8 \sum_{i=1}^n n = 8n^2 \in O(n^2)$$

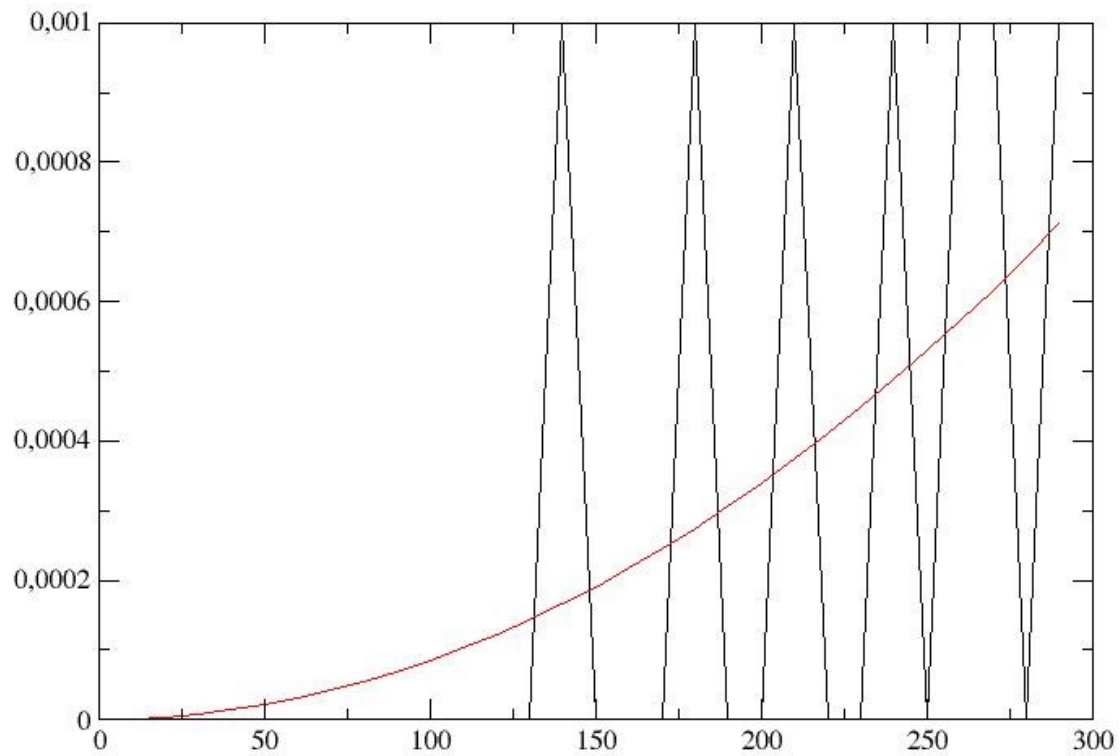
Cálculo de la eficiencia empírica:

La macro sigue ejecutándose 300 veces.



Cálculo de la eficiencia híbrida:

Esta sería la eficiencia híbrida de mi algoritmo.



Macro:

El contenido de mi macro para los dos ejercicios sería:

```
#!/bin/csh -vx
```

```
@ i = 10
```

```
echo ""> rell.dat
```

```
while ( $i < 300)
```

```
echo " $i `./rell $i`" >> rell.dat
```

```
@ i += 10
```

```
end
```