

# Ejercicio 1: Ordenación de la burbuja .

**El siguiente código realiza la ordenación mediante el algoritmo de la burbuja:**

**void ordenar(int \*v, int n) { for (int i=0; i<n-1; i++)**

**for (int j=0; j<n-i-1; j++) if (v[j]>v[j+1]) {**

**int aux = v[j]; v[j] = v[j+1]; v[j+1] = aux;**

**}**

**}**

**Calcule la eficiencia teórica de este algoritmo. A continuación replique el experimento que se ha hecho antes (búsqueda lineal) con este nuevo código. Debe:**

* **Crear un fichero ordenacion.cpp con el programa completo para realizar una ejecución del algoritmo.**
* **Crear un script ejecuciones\_ordenacion.csh en C-Shell que permite ejecutar varias veces el programa anterior y generar un fichero con los datos obtenidos.**
* **Usar gnuplot para dibujar los datos obtenidos en el apartado previo.**

**Los datos deben contener tiempos de ejecución para tamaños del vector 100, 600, 1100, ..., 30000.**

**Pruebe a dibujar superpuestas la función con la eficiencia teórica y la empírica. ¿Qué sucede?**

Eficiencia teorica:

void ordenar(int \*v, int n) {

n-2 n-2 n-2 n-2

for (int i=0; i<n-1; i++) ←T(n)= ∑ n-i-1 =∑ n - ∑ i - ∑ 1 =

i=0 i=0 i=0 i=0

= n(n-1)- n((n-1)/2)-(n-1) = n²-n -(n²/2)+(n/2)-n+1=

= (n²-3n+2)/2 ∈ **O(n²)**

n-i-2

for (int j=0; j<n-i-1; j++) ←∑ 1= n-i-1

j=0

if (v[j]>v[j+1]) { ←El if y lo que hay dentro vale O(1) int aux = v[j];

v[j] = v[j+1]; v[j+1] = aux;

}

}

Puesto que T(n)=(n²-3n+2)/2 ∈ O(n²) podemos afirmar que el orden de eficiencia del algoritmo de ordenación por burbuja es O(n²).

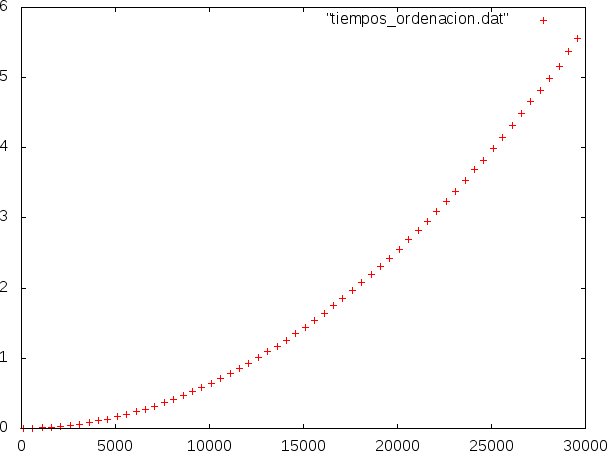
Eficiencia empírica:

Para medir el tiempo de ejecución del algoritmo de ordenación por burbuja, generamos un vector ordenado de mayor a menor , provocando de esta forma que se dé el peor caso posible. El programa tiene un argumentos que se le suministra en la línea de órdenes, el tamaño del vector.

* He creado el fichero **ordenacion.cpp** para realizar la ejecución del algoritmo.
* He creado el script ejecuciones\_ordenacion.csh para ejecutar varias veces el programa anterior para tamaños del vector 100, 600, 1100, ..., 30000 y generar un fichero con los datos obtenidos.

**-**He usado gnuplot para dibujar los datos obtenidos en el apartado previo, resultando:

# gnuplot> plot "tiempos\_ordenacion.dat"



Dibujar superpuestas la función con la eficiencia teórica y la empírica:

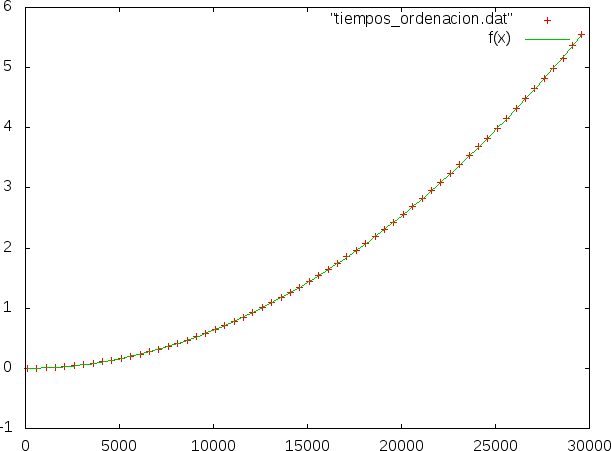
Para el calculo del ajuste teorico con el empirico primero he declarado la funcion f(x) como “f(x)=a\*x\*\*2+b\*x+c” ya que para el caso peor es 0(n²) .

Una vez que he declarado la funcion he pasado a ajustar con la orden “fit” para calcula la a, b y c. Despues he calculado el ajuste dibujando las dos graficas.

Pasos:

# gnuplot> f(x)=a\*x\*\*2+b\*x+c

1. **fit f(x) "tiempos\_ordenacion.dat" via a,b,c**
2. **gnuplot> plot "tiempos\_ordenacion.dat" , f(x)**



La grafica de la eficiencia teorica se ajusta con la de eficiencia empírica.