

# ESTRUCTURA DE DATOS

## *Práctica 1. Eficiencia de algoritmos*

Doble Grado de Informática y Matemáticas

Víctor Castro Serrano  
Maximino Suárez van Gelderen

28 de septiembre de 2017

### Condiciones de ejecución.

Dado que en los siguientes ejercicios hablaremos de la eficiencia de distintos programas, conviene detallar las condiciones en las que se han llevado a cabo las pruebas.

**Hardware:** Asus GL552VW, Intel Core i5-6300HQ CPU @ 2.30GHz 4 cores, Intel HD Graphics 530 (Skylake GT2), 12GB RAM.

**Sistema Operativo:** Ubuntu 16.04.3 LTS 64-bit.

**Compilador:** g++

**Opciones de compilación:** -g -o

### Ejercicio 5.

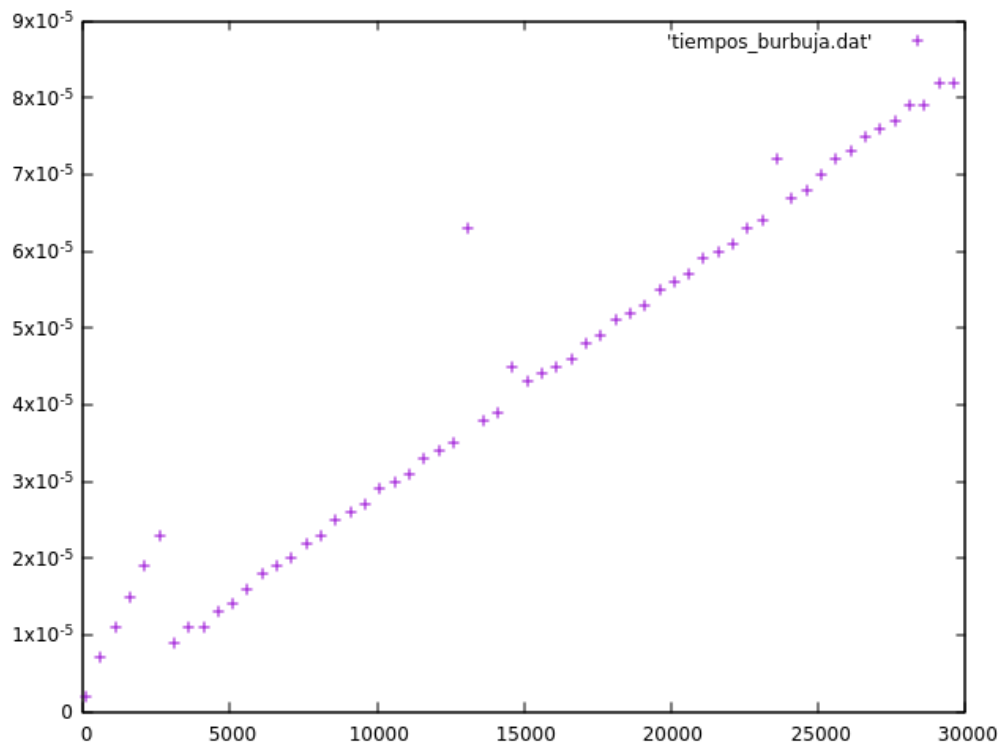
En este ejercicio utilizamos la ordeación por burbuja con una implementación ligeramente diferente **burbuja**

```
void ordenar(int *v, int n) {
    bool cambio=true;
    for (int i=0; i<n-1 && cambio; i++) {
        cambio=false;
        for (int j=0; j<n-i-1; j++)
            if (v[j]>v[j+1]) {
                cambio=true;
                int aux = v[j];
                v[j] = v[j+1];
                v[j+1] = aux;
            }
    }
}
```

El cambio consiste en que se ha introducido una nueva variable que permite saber en cada iteración del bucle externo si se ha producido un cambio en el vector. Si no se produce un cambio, el vector ya estará ordenado, por lo que no es necesario continuar ejecutando el código.

Si realizamos un estudio teórico de la eficiencia en el mejor caso, es decir, cuando el vector de entrada está ya ordenado, nos damos cuenta rápidamente de que el orden de eficiencia es lineal, es decir,  $T(n) \in O(n)$ . Esto es así porque tan sólo habría que recorrer una vez  $n - 1$  elementos del vector, hasta darse cuenta de que está ordenado.

Analicemos ahora la eficiencia empírica de este algoritmo en el caso mejor. Veamos la gráfica asociada:



Al realizarlo empíricamente, aumenta ligeramente el tiempo de ejecución conforme aumenta el tamaño del vector, por lo que tarda más en llenarlo, pero no es culpa del algoritmo de burbuja en sí el aumento en el tiempo de ejecución.