

# Práctica 3

Juan Llinares Mauri  
jlm109@alu.ua.es

## 1. Práctica 3 - Complejidad temporal: Cálculo analítico

### 1.1. Ejercicio 3

```
1 void exercise3 (vector <int> &v) {  
2     int i=1, n=v.size();  
3     bool swaped=true;  
4     while (swaped) {  
5         swaped=false;  
6         for (int j=n-1; j>=i; j--) {  
7             if (v[j] < v[j-1]) {  
8                 int x=v[j];  
9                 v[j]=v[j-1];  
10                v[j-1]=x;  
11                swaped=true;  
12            }  
13        }  
14        i++;  
15    }  
16 }
```

En este ejercicio contamos con un tamaño de problema  $n = v.size()$ .

Se trata de un programa que ordena de menor a mayor un vector dado desde el último número al primero. No obstante, si no hay ningún cambio durante la última pasada, el programa parará pues el vector ya estará ordenado. Es un algoritmo basado en *bubble sort* pero con la mejora de la variable *swaped*.

Aquí podemos diferenciar dos casos claros:

- **Peor caso:** Cuando el vector está ordenado a la inversa, es decir, de mayor a menor.
- **Mejor caso:** Cuando el vector ya está ordenado de menor a mayor.

**Peor caso** En este caso el bucle `for` será recorrido  $n*n$  veces, pues la variable `swaped` nunca será `false` hasta que el bucle ya no pueda realizar más intercambios. La complejidad de este caso será entonces de  $\Theta(n^2)$ , igual que la de *Bubble Sort* en su peor caso.

Para comprenderlo mejor, miramos el número de iteraciones totales que se realizan en este caso. Estas siguen la siguiente suma aritmética (dentro del bucle `for`):

$$(n-1) + (n-2) + (n-3) + \dots + 1 = \frac{n * (n-1)}{2} = \frac{n^2 - n}{2}$$

Como todas las demás operaciones fuera del bucle `for` son inmediatas (de coste  $O(1)$ ), podemos confirmar que el peor caso tiene complejidad temporal de  $\Theta(n^2)$ .

**Mejor caso** En este caso únicamente se recorrerá el bucle `for` una vez. Como no habrá cambio en la variable `swaped`, saltará al `while` al terminar las  $j = n-1$  veces que se ejecuta y terminará el programa. Por lo tanto, este caso tendrá una complejidad temporal de  $\Omega(n)$ .