

# **ALGORÍTMICA**

## **PRÁCTICA 2: DIVIDE Y VENCERÁS**



### **AUTORES:**

**Pablo Moreno Megías, Diego Lerena García, Manuel Vallejo Felipe, Ángel Díaz de la Torre,  
Francisco Navarro Morales, Marcel Kemp Muñoz y David Redondo Correa**

Algorítmica [PRACTICAS]  
Segundo curso del Grado de Ingeniería Informática.  
Universidad de Granada.  
curso 2016-2017.

# Índice

<b>1. Algoritmo Obvio. Fuerza Bruta.</b>	<b>2</b>
1.1. Algoritmo . . . . .	2
1.2. Eficiencia teórica. . . . .	2
1.3. Eficiencia empírica. . . . .	2
1.4. Eficiencia híbrida. . . . .	2
<b>2. Algoritmo Divide y Vencerás.</b>	<b>2</b>
2.1. Algoritmo . . . . .	2
2.2. Eficiencia teórica. . . . .	2
2.3. Eficiencia empírica. . . . .	2
2.4. Eficiencia híbrida. . . . .	2
<b>3. Comparación</b>	<b>3</b>

# 1 Algoritmo Obvio. Fuerza Bruta.

## 1.1. Algoritmo

## 1.2. Eficiencia teórica.

## 1.3. Eficiencia empírica.

## 1.4. Eficiencia híbrida.

# 2 Algoritmo Divide y Vencerás.

## 2.1. Algoritmo

Listing 1: A sample

```
1 #include<iostream>
2 #include<list>
3 using namespace std;

5 int main(int argc, char * argv[]){
    // primer argumento n de elementos
7     int inversiones = 0 ;
    list<int> V ; //original
9     list<int> M ; //mezclado

11    /*
        SE
13        RELLENAN
        LOS
15        VECTORES
        ...
17        ...
18    */
19
20    int inversiones = 0;
21    while(M.size() > 0){
22        int n = M[0] ;
23        inversiones += Posicion(V,0,V.size()-1,n) ; //O(log(n))
24        V.erase(n); // O(1)
25        M.erase(n); // O(1)
26    }
27
28    return inversiones ;
29 }
```

## 2.2. Eficiencia teórica.

## 2.3. Eficiencia empírica.

## 2.4. Eficiencia híbrida.

### **3 Comparación**