

# ITC-ADA-C1-2023: Assignment #1

Luis Ballado

luis.ballado@cinvestav.mx

CINVESTAV UNIDAD TAMAULIPAS — January 12, 2023

## 1 Diseñe un algoritmo para encontrar todos los elementos comunes en dos listas ordenadas de números



**Info:** Se hace la prueba con los siguientes valores  
2, 5, 5, 5 y 2, 2, 3, 5, 5, 7

### Pregunta 1

¿Cuál es el número máximo de comparaciones que realiza su algoritmo en función de las longitudes de las listas (m y n) respectivamente?  
Máximo de comparaciones

### 1.1 Pseudocodigo

---

#### Algorithm 1: Sum of Array Elements

---

```
entrada: Arreglo1, Arreglo2
salida : Arreglo3 (arreglo con repetidos)
CalcularLongitud(arreglo1)
CalcularLongitud(arreglo2)
for i ← 0 to longitud(arreglo1) do
    for j ← 0 to longitud(arreglo2) do
        if arreglo1[i] == arreglo2[j] then
            if !Existe(arr, arreglo1[i]) then
                agregar al arreglo de resultado a[i]
            break;
```

---

---

#### Algorithm 2: Función Existe

---

```
input : arreglo3, valor
output: verdadero ó falso (booleano si existe el elemento ó no)

Function Existe(arreglo3, valor)
    for i ← 0 to longitud(arreglo3) do
        if arreglo3[i] == valor then
            return Verdadero
    return Falso
```

---

## 1.2 Implementación

```
tarea1.cpp

1  #include <iostream>
2  #include <vector>
3
4  bool existe(std::vector<int> _vector_, int valor){
5      for(int i = 0; i < _vector_.size(); i++){ // n
6          if (_vector_[i] == valor){           // 1
7              return true;
8          }
9      }
10     return false;
11 }
12
13 //Complejidad funcion principal O(n^2)
14 int main(){
15
16     int a[] = {2,5,5,5};           // 1
17     int b[] = {2,2,3,5,5,7};       // 1
18
19     //vector de resultados
20     std::vector<int>arr;           // 1
21
22     //longitud array1
23     int a_length = sizeof(a)/sizeof(a[0]); // 1
24     //longitud array2
25     int b_length = sizeof(b)/sizeof(b[0]); // 1
26
27     //Usando fuerza bruta           // O(n^2)
28     for (int i = 0; i < a_length; i++){
29         for (int j = 0; j < b_length; j++){
30             if (a[i] == b[j]){      // 1
31                 //revisar si ya existe en el array
32                 if (!existe(arr,a[i])){ // 1
33                     arr.push_back(a[i]);
34                 }
35                 break;
36             }
37         }
38     }
39
40     // Imprimir resultado
41     std::cout << "El resultado es: \n"; // 1
42     for(int i = 0; i<arr.size(); i++){   // n
43         std::cout << arr[i] << "\n";    // 1
44     }
45 }
```

Ejecutar desde una terminal

Command Line

```
$ g++ -o ./tarea1 ./tarea1.cpp
$ ./tarea1
```

### 1.3 Análisis de complejidad del mejor y peor caso



MEJOR CASO: Explicar aqui



PEOR CASO: Explicar aqui