

# ITC-ADA-C1-2023: Assignment #3

Luis Ballado

luis.ballado@cinvestav.mx

CINVESTAV UNIDAD TAMAULIPAS — February 5, 2023

## 1 Considerando el algoritmo recursivo mostrado a continuación, responda las siguientes preguntas

---

### Algorithm 1: Algoritmo Misterio

---

```
entrada: Un arreglo  $A[0..n-1]$  de números reales
if  $n=1$  then
   $\perp$  return  $A[0]$ ;
else
   $temp \leftarrow \text{Misterio}(A[0..n-2])$ 
  if  $temp \leq A[n-1]$  then
     $\perp$  return  $temp$ ;
  else
     $\perp$  return  $A[n-1]$ ;
```

---

#### Pregunta 1

¿Qué calcula el algoritmo?

#### Pregunta 2

¿Cuál es el parámetro que indica el tamaño de la entrada del algoritmo?

#### Pregunta 3

¿Cuál es el parámetro que indica el tamaño de la entrada del algoritmo?

#### Pregunta 4

¿Cuál es el parámetro que indica el tamaño de la entrada del algoritmo?

Pregunta 5

¿Cuál es el parámetro que indica el tamaño de la entrada del algoritmo?

Pregunta 6

¿Cuál es el parámetro que indica el tamaño de la entrada del algoritmo?

Pregunta 7

¿Cuál es el parámetro que indica el tamaño de la entrada del algoritmo?

**2 Dado el problema de encontrar el determinante de una matriz A de  $n \times n$  , desarrolle los siguiente puntos:**

Pregunta 8

**Programa las versiones iterativa y recursiva del algoritmo para resolver el problema**

## **2.1 Implementación**

#### tarea1.cpp

```
1  #include <iostream>
2  #include <vector>
3
4  //Complejidad funcion principal 0(n^2)
5  //por el doble for que recorre los arreglos
6  int main(){
7
8      std::vector<int> a;
9      a.push_back(2);
10     a.push_back(5);
11     a.push_back(5);
12     a.push_back(5);
13
14     std::vector<int> b;
15     b.push_back(2);
16     b.push_back(2);
17     b.push_back(3);
18     b.push_back(5);
19     b.push_back(5);
20     b.push_back(7);
21
22     //vector de resultados
23     std::vector<int> arr;
24
25     int last_index = 0;
26
27     //Usando fuerza bruta // 0(n^2)
28     for (int i = 0; i < a.size(); i++){ // 0(n)
29         for (int j = i; j < b.size(); j++){ // 0(n)
30
31             //std::cout << a[i] << "<-a comparacion b->"<< b[
32             last_index] << "\n";
33
34             if (a[i] == b[last_index]){
35                 arr.push_back(a[i]);
36                 break; // romper ciclo cuando sean iguales
37             }
38
39             last_index = i+1; //indice auxiliar para avanzar
40         }
41     }
42
43     //-----
44     // Imprimir resultado
45     //-----
46
47     std::cout << "El resultado es: \n"; // 1
48     for(int i = 0; i<arr.size(); i++){ // n
49         std::cout << arr[i] << "\n";
50     }
51 }
52 }
```

ver código en github

Ejecutar desde una terminal

#### Command Line

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

Pregunta 9

**Analice matemáticamente cada versión el algoritmo por separado usando las metodologías vistas en clase. *Respuesta aquí***

Pregunta 10

**En base a los resultados obtenidos en el punto anterior determine cuál de los dos algoritmos es más eficiente *Respuesta aquí***

Pregunta 11

**Para cada uno de los dos algoritmos desarrollados aplique el método descrito en el apartado "Doubling ratio experiments" del libro Algorithms de Sedgewick y Wayne, generando instancias de tamaños 1000,2000,etc; hasta lograr un radio de  $2^b$ , ejecutando 20 pruebas con cada tamaño y con cada algoritmo. Registre sus resultados. *Respuesta aquí***

Pregunta 12

**Para cada algoritmo comparado realice una tabla con 5 predicciones, posteriores al tamaño con que se logró obtener el radio  $2^b$  *Respuesta aquí***

Pregunta 13

**Para cada algoritmo comparado grafique los siguientes resultados de sus ejecuciones *Respuesta aquí***

Pregunta 14

**Con base en los experimentos realizados y considerando un tiempo máximo de ejecución sobre su computadora de 7 días, ¿Cuál es el tamaño máximo de entrada que puede resolver cada algoritmo analizado? *Respuesta aquí***

Pregunta 15

**Conclusiones respecto al orden de crecimiento de cada algoritmo observado empíricamente y constrúelas contra los resultados de sus análisis matemático *Respuesta aquí***