

ANALISIS Y DISEÑO DE ALGORITMOS I Periodo I - 2023

Jesús Aranda

Universidad del Valle
Escuela de Ingeniería de Sistemas y Computación

Este documento es una adaptación del material original del profesor Oscar Bedoya



Lo que sabemos

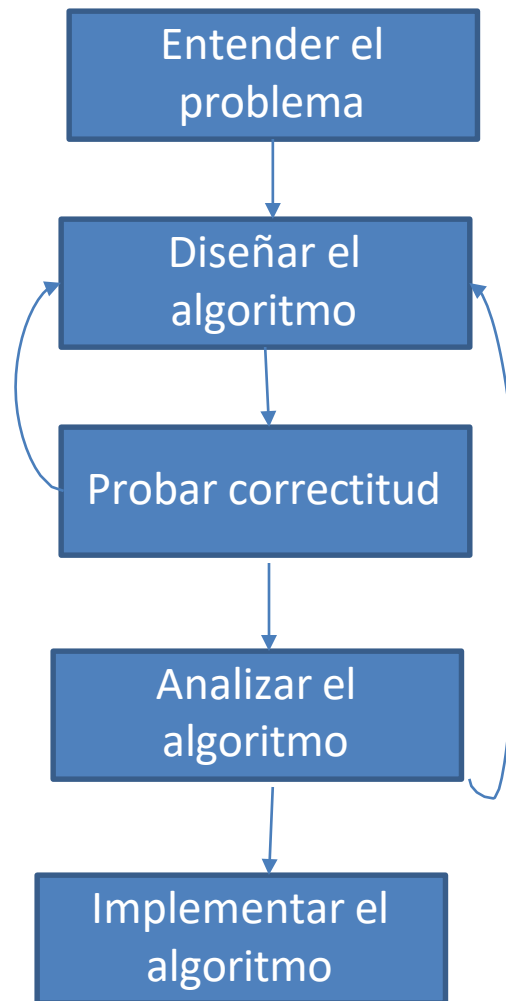
Un algoritmo es un **procedimiento computacional** que toma un valor o conjunto de valores como entrada y produce un valor o conjunto de valores como salida.

Un algoritmo es una secuencia de instrucciones precisas (no ambiguas) para resolver un problema.

- El rango de entradas del algoritmo debe ser definido cuidadosamente.
- El mismo algoritmo puede ser implementado de diferentes formas
- Pueden existir diferentes algoritmos para el mismo problema.
- Algoritmos para el mismo problema pueden ser muy diferentes con respecto a la eficiencia.

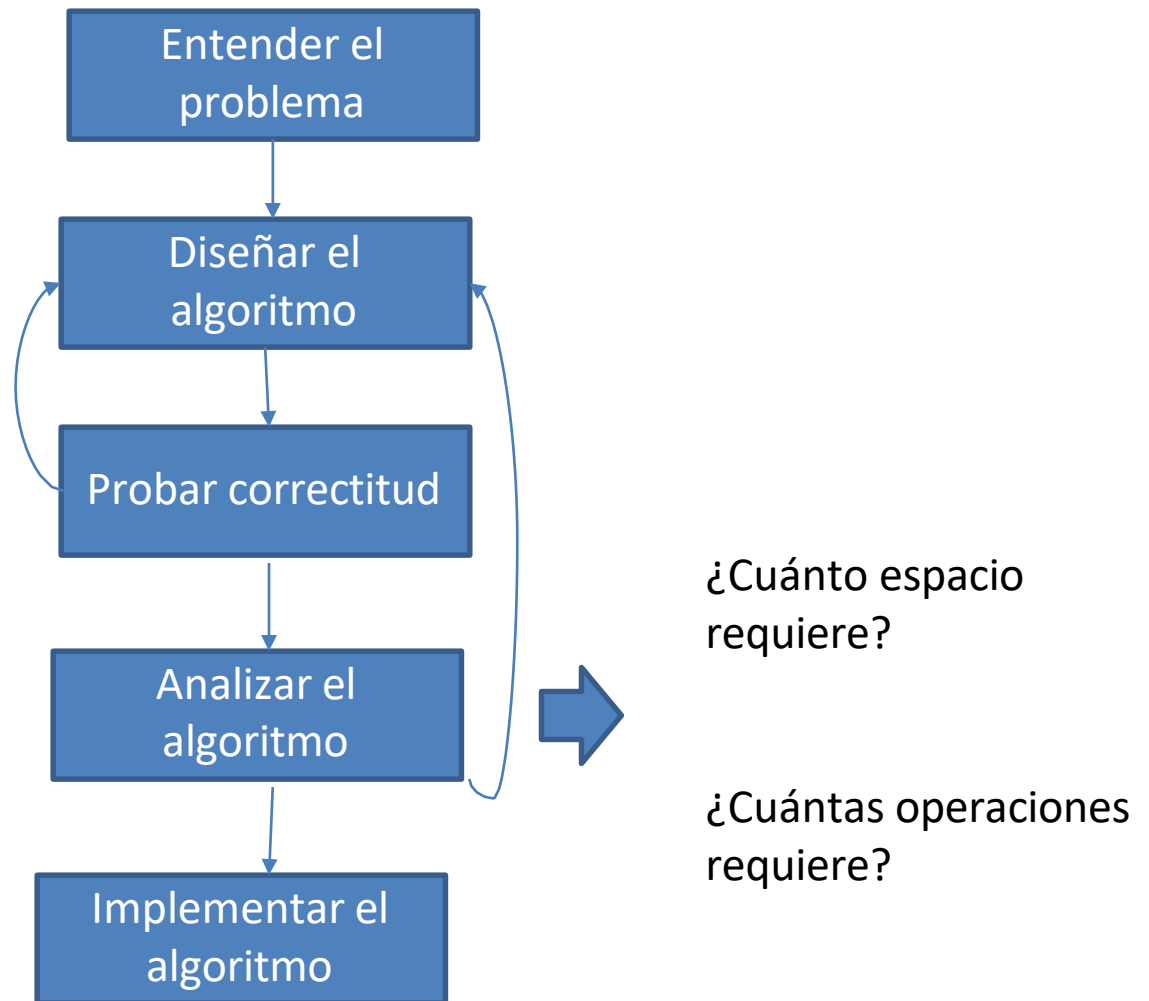
Lo que sabemos

Diseño y Análisis de Algoritmos



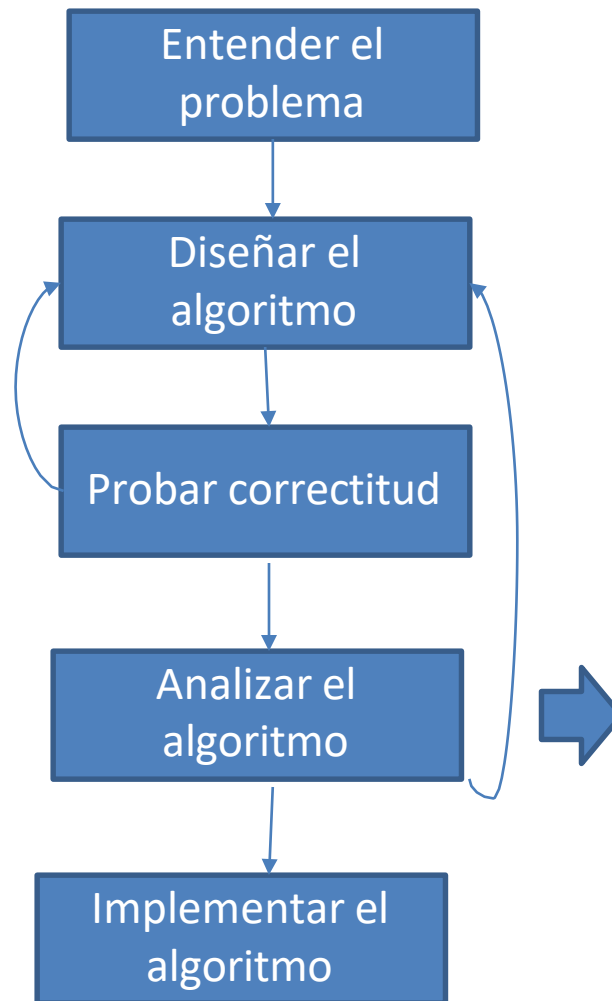
Lo que sabemos

Diseño y Análisis de Algoritmos



Lo que sabemos

Análisis de Algoritmos



¿Cuánto espacio requiere?

¿Cuántas
operaciones
requiere?

Pasos para el análisis de eficiencia de un algoritmo:



Determinar el (los) parámetro(s) que indican el tamaño de la entrada



Identificar la operación básica del algoritmo



Verificar si el número de veces que se ejecuta la operación básica depende exclusivamente del tamaño de la entrada. En caso contrario, se tendrán que analizar el peor (mejor) caso o el caso promedio.



Definir una expresión matemática que represente el número de veces que la operación básica se ejecuta en el algoritmo.



Determinar una solución explícita para la expresión del punto anterior, y determinar su orden de crecimiento.

Pasos para el análisis de eficiencia de un algoritmo:

Pasos para el análisis de eficiencia de un algoritmo.

Ejemplo

```
Algoritmo MaximoElemento (A[0 .. n-1])
// Determina el valor del elemento más grande
// de un arreglo
// Entrada: Un arreglo A[0..n-1] de números
// Salida: El valor del elemento más grande de A
    maxval ← A[0]
    for i ← 1 to n-1 do
        if A[i] > maxval
            maxval ← A[i]
    return maxval
```

Tamaño de entrada	n, indicando el # de elementos del arreglo
Operación Básica	Línea 3
Depende exclusivamente del tamaño de entrada	Sí
Expresión correspondiente al # veces que se ejecuta la operación básica	$\sum_{i=1}^{n-1} 1$
Forma explícita de la expresión	n-1
Orden de crecimiento	O(n)

Pasos para el análisis de eficiencia de un algoritmo:

Pasos para el análisis de eficiencia de un algoritmo.

Realizar ejercicio 01 y ejercicio 02 en clase

Algoritmos en la computación

Problema 1: ???

Entrada: $n \in \mathbb{Z}^+$

Salida: 1, si $n=2$

0, si $n=1$

1, si $n > 2 \wedge n \bmod i \neq 0 \ \forall i \in \{2, \dots, n-1\}$

0, si $n > 2 \wedge \exists x \mid n \bmod x = 0 \wedge x \in \{2, \dots, n-1\}$

Algoritmos en la computación

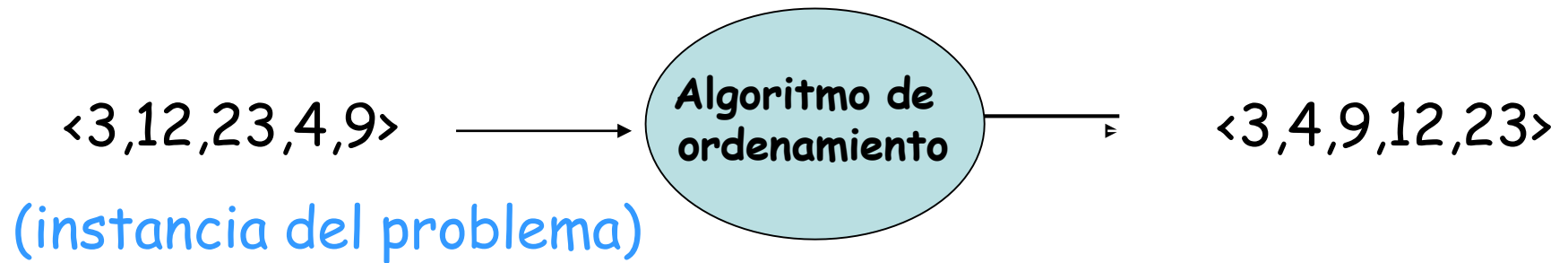
Problema 2: ???

Entrada: $S = \langle a_1, a_2, \dots, a_n \rangle$

Salida: una permutación de S , $S' = \langle a_1', a_2', \dots, a_n' \rangle$ tal que
 $a_1' < a_2' < \dots < a_n'$

Algoritmos en la computación

Instancia de un problema

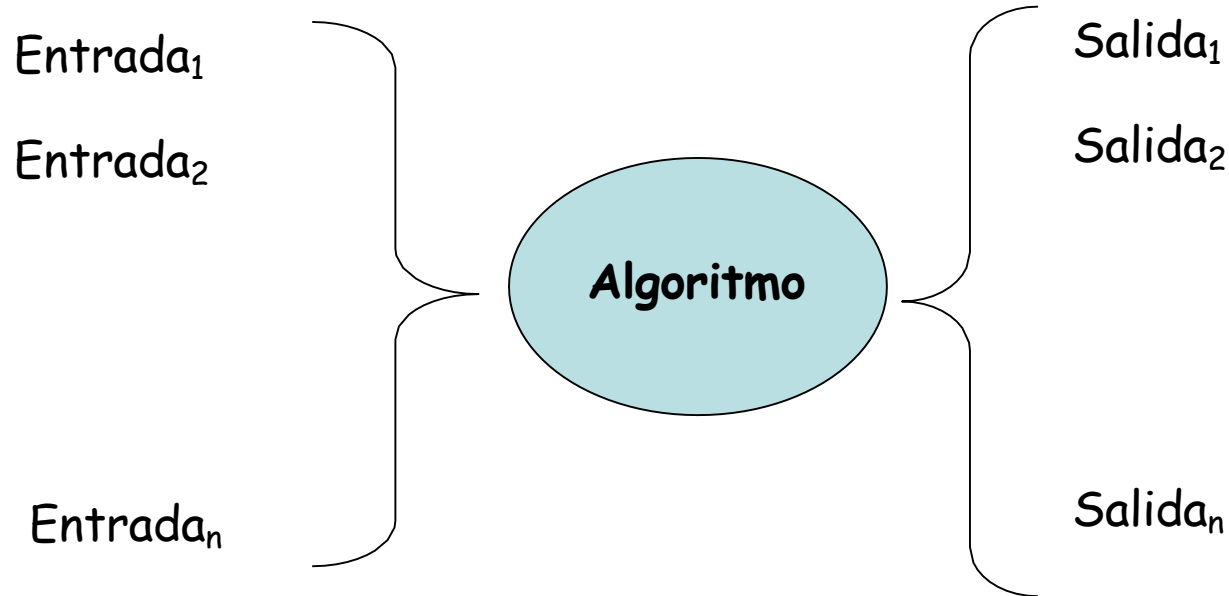


Una instancia es una entrada válida para el algoritmo

Algoritmos en la computación

Correctitud

Se dice que un algoritmo es **correcto**, si para cada *instancia*, el algoritmo termina con la salida correcta



Algoritmos en la computación

```
primo(int n){  
    if (n==1)  
        return 0;  
  
    if (n%2==0)  
        return 0;  
  
    else  
  
        return 1;  
  
}
```

Para cada instancia, el algoritmo termina con la salida correcta

¿Es el algoritmo primo correcto?

Algoritmos en la computación

```
primo(int n){  
    if (n==1)  
        return 0;  
    else{  
  
        int c=0;  
        for (int i=2; i<n; i++)  
            if (n%i==0) c++;  
        if (c==0)  
            return 1;  
        else return 0;  
    }  
}
```

Para cada instancia, el algoritmo termina con la salida correcta

¿Es el algoritmo primo correcto?

Algoritmos en la computación

```
primo(int n){  
    if (n==1)  
        return 0;  
    else{  
  
        int c=0;  
        for (int i=3; i<n; i++)  
            if (n%i==0) c++;  
  
        if (c==0)  
            return 1;  
        else return 0;  
    }  
}
```

Para cada instancia, el algoritmo termina con la salida correcta

¿Es el algoritmo primo correcto?

Algoritmos en la computación

Tipos de problemas solucionados utilizando algoritmos eficientes:

- **Bioinformática**: Identificar genes en secuencias de ADN (genoma humano 3200000000pb). Se almacena la información observada hasta el momento en bases de datos y se analizan con algoritmos que debe ser eficientes
- **Búsquedas en Internet**: Dada la cantidad de información indexada, responder de forma correcta la solicitud de una búsqueda en Internet

Algoritmos en la computación

Tipos de problemas solucionados utilizando algoritmos eficientes:

- **Tratamiento de colisiones de objetos:** Detectar una colisión entre dos cuerpos en un espacio 3D
- **Búsquedas sobre videos:** En un biblioteca, un usuario desea encontrar todos los videos según una etiqueta

Algoritmos en la computación

Análisis de algoritmos

Meta: Comparar algoritmos que resuelven un mismo problema

- Correctitud
- Eficiencia
 - Tiempo
 - Espacio
- Estructuras de datos utilizadas
- Modelo computacional
- El tipo y número de datos con los cuales trabaja (escalabilidad)