

# Introducción a los Algoritmos

Miguel Raggi

Algoritmos

Escuela Nacional de Estudios Superiores  
UNAM

5 de febrero de 2019

# Índice:

## 1 Algoritmos

- ¿Qué son?
- Componentes
- Pseudo-Código
- Utilidad

# Índice:

## 1 Algoritmos

- ¿Qué son?
- Componentes
- Pseudo-Código
- Utilidad

# ¿Qué es un Algoritmo?

## Definición

Un **algoritmo** es una descripción matemática de un procedimiento lógico que puede seguir una computadora.

# ¿Qué es un Algoritmo?

## Definición

Un **algoritmo** es una descripción matemática de un procedimiento lógico que puede seguir una computadora.

- Es como una receta de cocina, o una serie de instrucciones, que te dicen qué hacer en cada momento.

# ¿Qué es un Algoritmo?

## Definición

Un **algoritmo** es una descripción matemática de un procedimiento lógico que puede seguir una computadora.

- Es como una receta de cocina, o una serie de instrucciones, que te dicen qué hacer en cada momento.
- **Ejemplos.** Hay algoritmos para:

# ¿Qué es un Algoritmo?

## Definición

Un **algoritmo** es una descripción matemática de un procedimiento lógico que puede seguir una computadora.

- Es como una receta de cocina, o una serie de instrucciones, que te dicen qué hacer en cada momento.
- **Ejemplos.** Hay algoritmos para:
  - Sumar, restar, multiplicar, dividir dos números.

# ¿Qué es un Algoritmo?

## Definición

Un **algoritmo** es una descripción matemática de un procedimiento lógico que puede seguir una computadora.

- Es como una receta de cocina, o una serie de instrucciones, que te dicen qué hacer en cada momento.
- **Ejemplos.** Hay algoritmos para:
  - Sumar, restar, multiplicar, dividir dos números.
  - Ordenar una lista de números.



# ¿Qué es un Algoritmo?

## Definición

Un **algoritmo** es una descripción matemática de un procedimiento lógico que puede seguir una computadora.

- Es como una receta de cocina, o una serie de instrucciones, que te dicen qué hacer en cada momento.
- **Ejemplos.** Hay algoritmos para:
  - Sumar, restar, multiplicar, dividir dos números.
  - Ordenar una lista de números.
  - Encontrar el  $n$ -simo número primo.

# ¿Qué es un Algoritmo?

## Definición

Un **algoritmo** es una descripción matemática de un procedimiento lógico que puede seguir una computadora.

- Es como una receta de cocina, o una serie de instrucciones, que te dicen qué hacer en cada momento.
- **Ejemplos.** Hay algoritmos para:
  - Sumar, restar, multiplicar, dividir dos números.
  - Ordenar una lista de números.
  - Encontrar el  $n$ -simo número primo.
  - Encontrar el camino más corto entre dos puntos de una ciudad.

# ¿Qué es un Algoritmo?

## Definición

Un **algoritmo** es una descripción matemática de un procedimiento lógico que puede seguir una computadora.

- Es como una receta de cocina, o una serie de instrucciones, que te dicen qué hacer en cada momento.
- **Ejemplos.** Hay algoritmos para:
  - Sumar, restar, multiplicar, dividir dos números.
  - Ordenar una lista de números.
  - Encontrar el  $n$ -simo número primo.
  - Encontrar el camino más corto entre dos puntos de una ciudad.
  - Encontrar donde aparece por primera vez una palabra dentro de un texto.

# ¿Cuáles son las partes de un algoritmo?

# ¿Cuáles son las partes de un algoritmo?

- Un algoritmo usualmente tiene **entradas** y **salidas**.

# ¿Cuáles son las partes de un algoritmo?

- Un algoritmo usualmente tiene **entradas** y **salidas**.
- Las “entradas” son la información en donde el algoritmo corre. Las “salidas” son el resultado.

# ¿Cuáles son las partes de un algoritmo?

- Un algoritmo usualmente tiene **entradas** y **salidas**.
- Las “entradas” son la información en donde el algoritmo corre. Las “salidas” son el resultado.
- Hay descripciones de **alto nivel** y de **bajo nivel**

# ¿Cuáles son las partes de un algoritmo?

- Un algoritmo usualmente tiene **entradas** y **salidas**.
- Las “entradas” son la información en donde el algoritmo corre. Las “salidas” son el resultado.
- Hay descripciones de **alto nivel** y de **bajo nivel**
- **Alto nivel**: Son descripciones donde dices a grandes rasgos qué hace el algoritmo.



# ¿Cuáles son las partes de un algoritmo?

- Un algoritmo usualmente tiene **entradas** y **salidas**.
- Las “entradas” son la información en donde el algoritmo corre. Las “salidas” son el resultado.
- Hay descripciones de **alto nivel** y de **bajo nivel**
- **Alto nivel**: Son descripciones donde dices a grandes rasgos qué hace el algoritmo.
- **Bajo nivel**: Descripciones muy específicas de qué hacer en cada momento. Usualmente en **pseudocódigo** (o código).

# Pseudo-Código

- Pseudo-código es una descripción de cómo programar un algoritmo, independientemente del lenguaje de programación que se quiera utilizar.

# Pseudo-Código

- Pseudo-código es una descripción de cómo programar un algoritmo, independientemente del lenguaje de programación que se quiera utilizar.
- Dentro del pseudocódigo podemos usar otros algoritmos o funciones.

# Pseudo-Código

- Pseudo-código es una descripción de cómo programar un algoritmo, independientemente del lenguaje de programación que se quiera utilizar.
- Dentro del pseudocódigo podemos usar otros algoritmos o funciones.
- Hay varios detalles particulares de cada lenguaje que podemos ignorar dentro del pseudocódigo.

# Pseudo-Código

- Pseudo-código es una descripción de cómo programar un algoritmo, independientemente del lenguaje de programación que se quiera utilizar.
- Dentro del pseudocódigo podemos usar otros algoritmos o funciones.
- Hay varios detalles particulares de cada lenguaje que podemos ignorar dentro del pseudocódigo.
- Usualmente, con un poco de práctica, una vez que se tiene el pseudocódigo, no es difícil programarlo.

# Pseudo-Código

- Pseudo-código es una descripción de cómo programar un algoritmo, independientemente del lenguaje de programación que se quiera utilizar.
- Dentro del pseudocódigo podemos usar otros algoritmos o funciones.
- Hay varios detalles particulares de cada lenguaje que podemos ignorar dentro del pseudocódigo.
- Usualmente, con un poco de práctica, una vez que se tiene el pseudocódigo, no es difícil programarlo.
- Usualmente.

# Algoritmo: Encontrar el número más grande

- **Entrada:** Una lista de números  $A$ .

# Algoritmo: Encontrar el número más grande

- **Entrada:** Una lista de números  $A$ .
- **Salida:** El elemento más grande de  $A$ .



# Algoritmo: Encontrar el número más grande

- **Entrada:** Una lista de números  $A$ .
- **Salida:** El elemento más grande de  $A$ .
- **MayorHastaAhora**  $:= -\infty$
- Para cada  $a \in A$ 
  - Si ( $a > \text{MayorHastaAhora}$ )
    - **MayorHastaAhora**  $:= a$
- **Regresa** **MayorHastaAhora**

# Código

El código implementado en python:

```
def EncontrarMayor(A):  
    MayorHastaAhora = A[0]  
    for a in A:  
        if (a > MayorHastaAhora):  
            MayorHastaAhora = a  
    return MayorHastaAhora
```

# ¿Para qué sirven los algoritmos?

# ¿Para qué sirven los algoritmos?

- Separa la idea de cómo resolver un problema de su implementación en algún lenguaje particular.

# ¿Para qué sirven los algoritmos?

- Separa la idea de cómo resolver un problema de su implementación en algún lenguaje particular.
- Se dice que la mitad del aumento de la velocidad en las computadoras en los últimos años es gracias a mejores algoritmos (la otra mitad es por mejor hardware).

# ¿Para qué sirven los algoritmos?

- Separa la idea de cómo resolver un problema de su implementación en algún lenguaje particular.
- Se dice que la mitad del aumento de la velocidad en las computadoras en los últimos años es gracias a mejores algoritmos (la otra mitad es por mejor hardware).
- ¿Por qué les sirven a ustedes?

# ¿Para qué sirven los algoritmos?

- Separa la idea de cómo resolver un problema de su implementación en algún lenguaje particular.
- Se dice que la mitad del aumento de la velocidad en las computadoras en los últimos años es gracias a mejores algoritmos (la otra mitad es por mejor hardware).
- ¿Por qué les sirven a ustedes?
- En realidad, la mayoría de los algoritmos que veremos, **ustedes nunca deberán programarlos en su vida profesional.**

# ¿Para qué sirven los algoritmos?

- Separa la idea de cómo resolver un problema de su implementación en algún lenguaje particular.
- Se dice que la mitad del aumento de la velocidad en las computadoras en los últimos años es gracias a mejores algoritmos (la otra mitad es por mejor hardware).
- ¿Por qué les sirven a ustedes?
- En realidad, la mayoría de los algoritmos que veremos, **ustedes nunca deberán programarlos en su vida profesional**.
- Por ejemplo, tú no debes hacer un programa para ordenar una lista de cosas. Si necesitas eso, debes utilizar uno hecho por profesionales.



# ¿Para qué sirven los algoritmos?

- Separa la idea de cómo resolver un problema de su implementación en algún lenguaje particular.
- Se dice que la mitad del aumento de la velocidad en las computadoras en los últimos años es gracias a mejores algoritmos (la otra mitad es por mejor hardware).
- ¿Por qué les sirven a ustedes?
- En realidad, la mayoría de los algoritmos que veremos, **ustedes nunca deberán programarlos en su vida profesional.**
- Por ejemplo, tú no debes hacer un programa para ordenar una lista de cosas. Si necesitas eso, debes utilizar uno hecho por profesionales.
- Pero te ayudan a tomar decisiones. Además, pues vemos cosas ya hechas y bien estudiadas primero, para que cuando tengan que hacer algo nuevo, entiendan las bases.

¿Qué significa que un algoritmo sea mejor que otro?

# ¿Qué significa que un algoritmo sea mejor que otro?

Pues para comparar dos algoritmos, primero que nada, deben hacer lo mismo. Es decir, con la misma entrada, deben regresar la misma salida.

# ¿Qué significa que un algoritmo sea mejor que otro?

Pues para comparar dos algoritmos, primero que nada, deben hacer lo mismo. Es decir, con la misma entrada, deben regresar la misma salida.

- Mayor velocidad.

# ¿Qué significa que un algoritmo sea mejor que otro?

Pues para comparar dos algoritmos, primero que nada, deben hacer lo mismo. Es decir, con la misma entrada, deben regresar la misma salida.

- Mayor velocidad.
- Use menos memoria/disco duro.

# ¿Qué significa que un algoritmo sea mejor que otro?

Pues para comparar dos algoritmos, primero que nada, deben hacer lo mismo. Es decir, con la misma entrada, deben regresar la misma salida.

- Mayor velocidad.
- Use menos memoria/disco duro.
- Más “elegantes” o más fáciles de entender.

# ¿Qué significa que un algoritmo sea mejor que otro?

Pues para comparar dos algoritmos, primero que nada, deben hacer lo mismo. Es decir, con la misma entrada, deben regresar la misma salida.

- Mayor velocidad.
- Use menos memoria/disco duro.
- Más “elegantes” o más fáciles de entender.
- Más fáciles de programar.

# ¿Qué significa que un algoritmo sea mejor que otro?

Pues para comparar dos algoritmos, primero que nada, deben hacer lo mismo. Es decir, con la misma entrada, deben regresar la misma salida.

- Mayor velocidad.
- Use menos memoria/disco duro.
- Más “elegantes” o más fáciles de entender.
- Más fáciles de programar.
- Si puede haber error, más **robusto** (es decir, que el error sea menor).