ANÁLISIS DE ALGORITMOSDIVIDE Y VENCERÁS

línea corta

Angel Eduardo Muñoz Perez

Luis Javier Garnica Escamilla

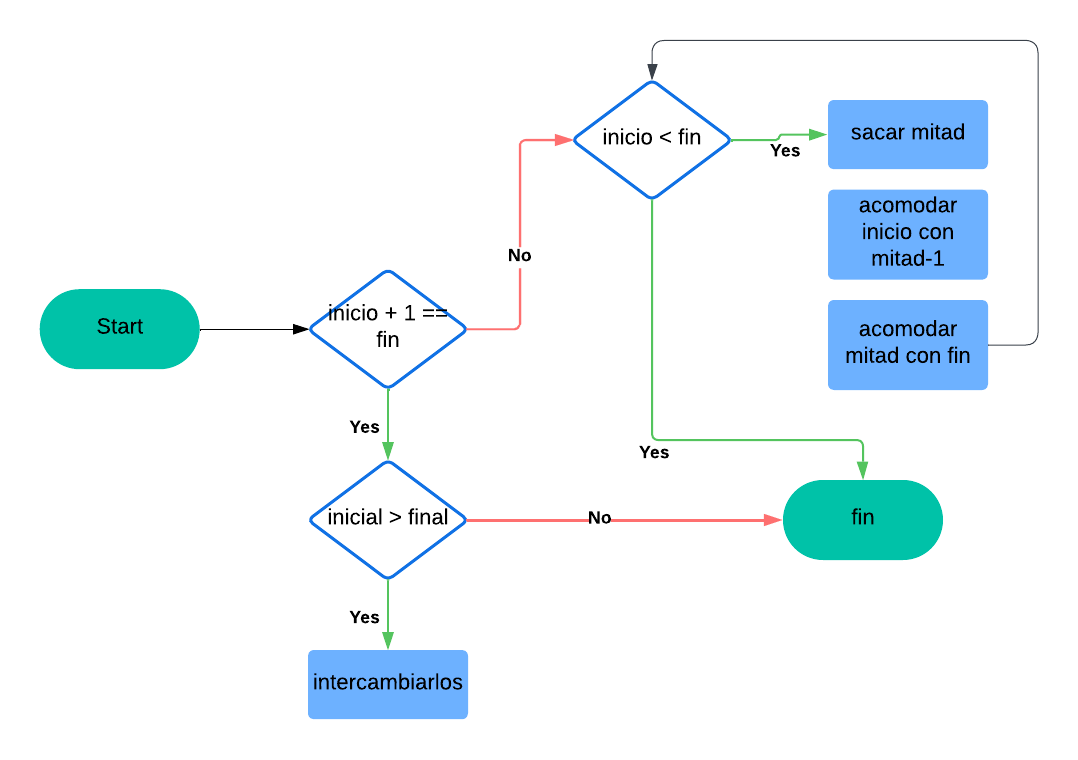
Fabián Emmanuel Candia Villa  
24 Enero 2025

# Introducción

Este es un ejemplo de MergeSort, este algoritmo se encarga de ir dividiendo un solo conjunto de datos en mitades cada vez más pequeñas para poder ordenarlas, y finalmente volverlas a juntar en una única lista ordenada de datos.

Cómo funciona:

Al empezar el algoritmo, revisamos si el primer elemento de la lista + 1 es igual a el último, en caso de ser así, revisamos entonces si es mayor, en caso de serlo, simplemente cambiamos ambos elementos de lugar. En caso contrario, simplemente acabamos.  
Si el primer elemento de la lista + 1 no es igual al último, revisamos entonces si el último elemento es mayor al primero, en caso de serlo, dividimos el array por mitad, ponemos el primer elemento un elemento antes que el que está en medio - 1, y ponemos el elemento del medio en el final.  
Comprobamos una vez más si el último elemento es mayor al primero, y si lo es, terminamos la ejecución del algoritmo.



Ejemplo de un código que realiza este algoritmo en python:

from random import random

"""

Ordena una lista utilizando el algoritmo de ordenación por mezcla (merge sort).

Args:

array (list): Lista de elementos a ordenar.

Returns:

list: Lista ordenada.

"""

def ordenar\_por\_mezcla(array):

if len(array) <= 1:

return array

mitad = len(array) // 2

izquierda = ordenar\_por\_mezcla(array[:mitad])

derecha = ordenar\_por\_mezcla(array[mitad:])

return mezclar(izquierda, derecha)

"""

Mezcla dos listas ordenadas en una sola lista ordenada.

Args:

izquierda (list): Primera lista ordenada.

derecha (list): Segunda lista ordenada.

Returns:

list: Lista combinada y ordenada.

"""

def mezclar(izquierda, derecha):

array\_ordenado = []

i = j = 0

while i < len(izquierda) and j < len(derecha):

if izquierda[i] < derecha[j]:

array\_ordenado.append(izquierda[i])

i += 1

else:

array\_ordenado.append(derecha[j])

j += 1

array\_ordenado.extend(izquierda[i:])

array\_ordenado.extend(derecha[j:])

return array\_ordenado

"""

Realiza una búsqueda binaria en una lista ordenada.

Args:

array\_entrada (list): Lista ordenada en la que se realizará la búsqueda.

busqueda (int): Elemento a buscar en la lista.

Returns:

tuple: Una tupla que contiene el elemento encontrado y un booleano indicando si fue encontrado.

Si el elemento no se encuentra, retorna "No encontrado".

"""

def binary\_search(array\_entrada, busqueda):

if len(array\_entrada) == 0:

return "No encontrado"

mitad = len(array\_entrada) // 2

if array\_entrada[mitad] == busqueda:

return busqueda, True

elif array\_entrada[mitad] < busqueda:

return binary\_search(array\_entrada[mitad + 1:], busqueda)

else:

return binary\_search(array\_entrada[:mitad], busqueda)

while True:

print("""\033[1m \033[33m

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .\_\_

/ \ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_ \_\_\_\_ / \_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_/ |\_ / \_ \ \\_\_\_\_\_\_ \/ \_\_\_\_\_/ \_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_ | |\_\_

/ \ / \\_/ \_\_ \\_ \_\_ \/ \_\_\_\\_/ \_\_ \ \\_\_\_\_\_ \ / \_ \\_ \_\_ \ \_\_\ > \_ </\ | | \_/\\_\_\_\_\_ \\_/ \_\_ \\\\\_\_ \\\\\_ \_\_ \\_/ \_\_\_\| | \\

/ Y \ \_\_\_/| | \/ /\_/ > \_\_\_/ / ( <\_> ) | \/| | / <\_\ \/ | | \/ \ \_\_\_/ / \_\_ \| | \/\ \\_\_\_| Y \\

\\_\_\_\_|\_\_ /\\_\_\_ >\_\_| \\_\_\_ / \\_\_\_ >\_\_\_\_\_\_\_ /\\_\_\_\_/|\_\_| |\_\_| \\_\_\_\_\_\ \ |\_\_\_\_\_\_ /\_\_\_\_\_\_\_ /\\_\_\_ >\_\_\_\_ /\_\_| \\_\_\_ >\_\_\_| /

\/ \/ /\_\_\_\_\_/ \/ \/ \/ \/ \/ \/ \/ \/ \/

\033[0m""")

array = []

n = int(input("Ingrese el tamaño del array: "))

for i in range(n):

array.append(int(random() \* 100))

print(f"Termino: {array[i]}")

print(f"Array desordenado: {array}")

array\_ordenado = ordenar\_por\_mezcla(array)

print(f"Array ordenado: {array\_ordenado}")

buscar = int(input("Ingrese el número a buscar: "))

print(f"Encontrado?: {binary\_search(array\_ordenado, buscar)}")

print("Deseas continuar? (s/n)")

continuar = input()

if continuar == "n":

break