

A vertical strip on the left side of the slide containing four distinct abstract geometric patterns in teal and white. From top to bottom: a circle split vertically; a four-pointed star with concave sides; a semi-circle split vertically; and a circle with concentric rings and a triangular grid pattern.

**DAY 5**

---

# **INTRO TO PROGRAMMING**

---

**BY: EVAN AURELRIUS**



# Timeline

- 1 SORTING
- 2 BUBBLE SORT
- 3 SELECTION SORT
- 4 INSERTION SORT
- 5 MERGE SORT



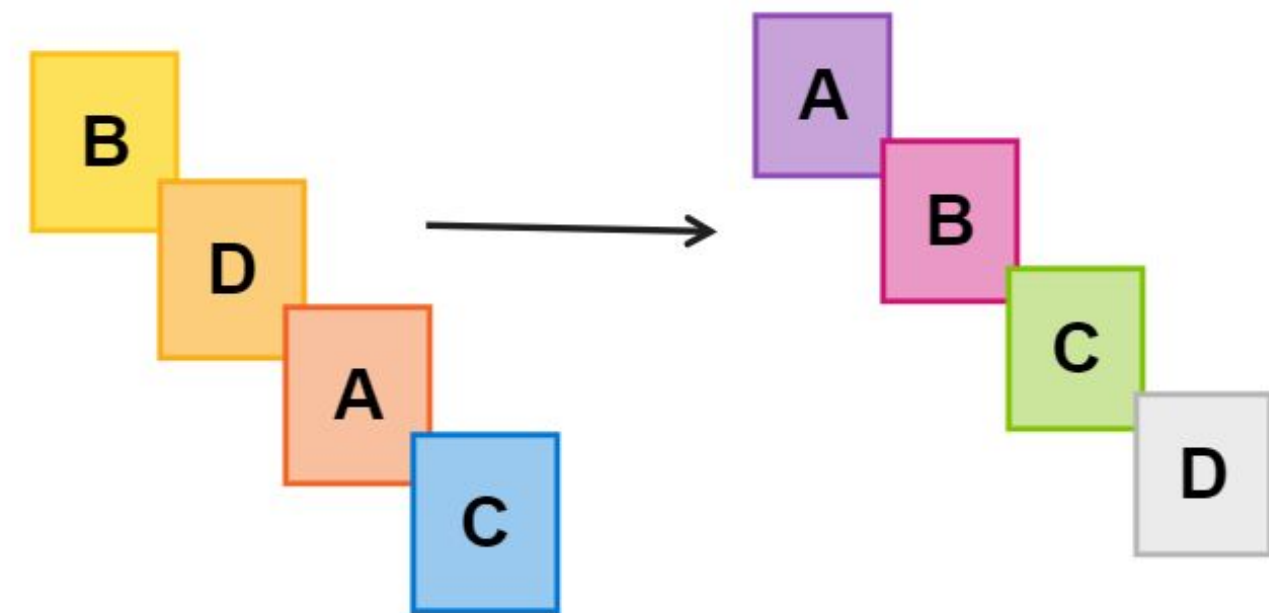
# SORTING



# SORTING

Sorting adalah proses mengurutkan elemen-elemen dalam suatu list atau array berdasarkan kriteria tertentu, seperti ukuran, nilai, atau urutan alfabet. Python menyediakan berbagai algoritma sorting yang dapat digunakan, seperti Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, dan Quick Sort.

## Sorting Algorithms



# BUBBLE SORT





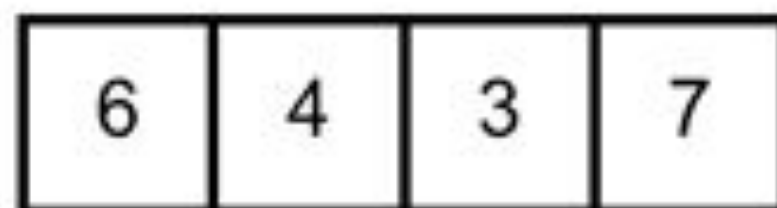
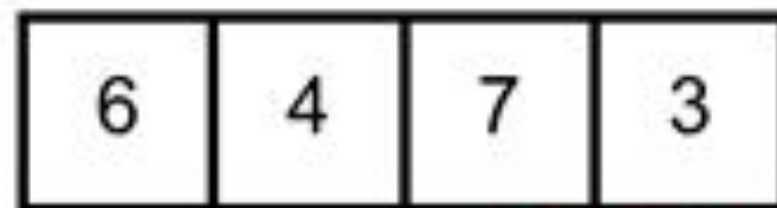
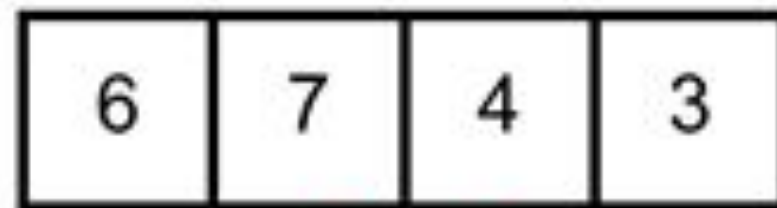
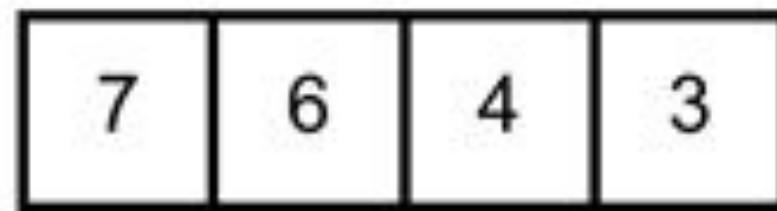
# BUBBLE SORT – (VISUALIZE)

**Bubble Sort** adalah salah satu algoritma sorting yang sederhana. Algoritma ini bekerja dengan cara membandingkan setiap pasangan elemen yang berurutan dalam daftar, dan menukar posisi dua elemen jika urutan mereka tidak sesuai dengan kriteria pengurutan.

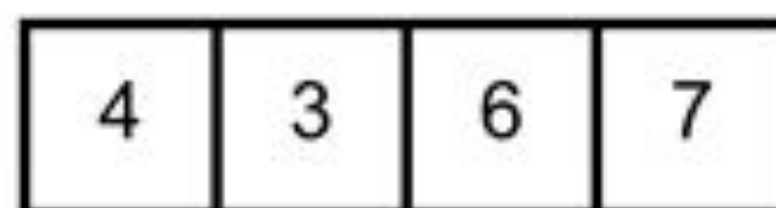
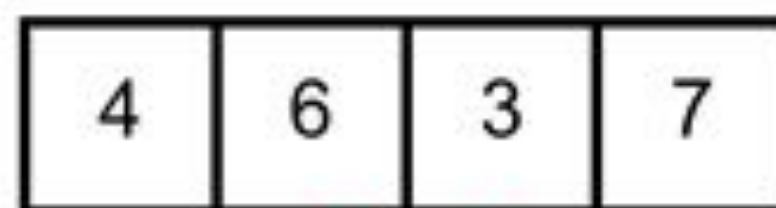
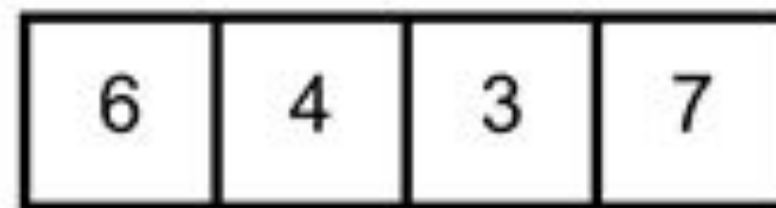
Algoritma Bubble Sort memiliki kompleksitas waktu  $O(n^2)$ , yang berarti semakin besar ukuran data, semakin lama waktu yang dibutuhkan untuk mengurutkan daftar tersebut.



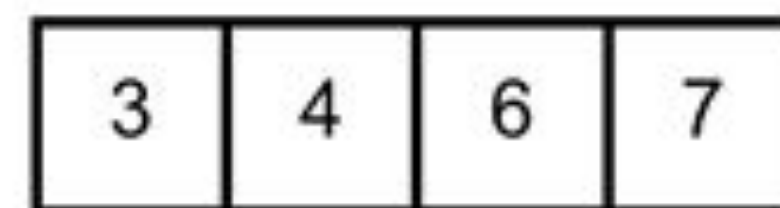
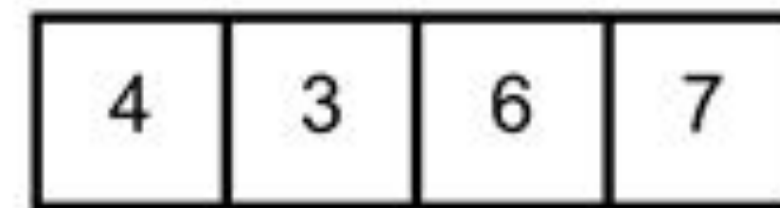
### First pass



### Second pass



### Third pass



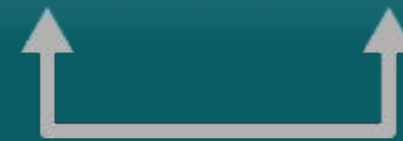
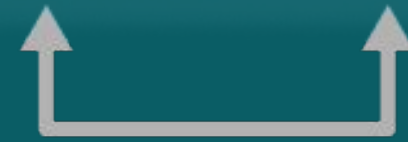
# SELECTION SORT





# Selection Sort – (Visualize)

- Selection Sort adalah algoritma sorting yang sederhana. Algoritma ini bekerja dengan cara mencari elemen terkecil dalam daftar, kemudian menukarkannya dengan elemen di posisi pertama. Setelah itu, algoritma ini akan mencari elemen terkecil berikutnya dan menukarkannya dengan elemen di posisi kedua, dan seterusnya.
- Algoritma Selection Sort memiliki kompleksitas waktu  $O(n^2)$ , sama halnya dengan Bubble Sort.



# INSERTION SORT

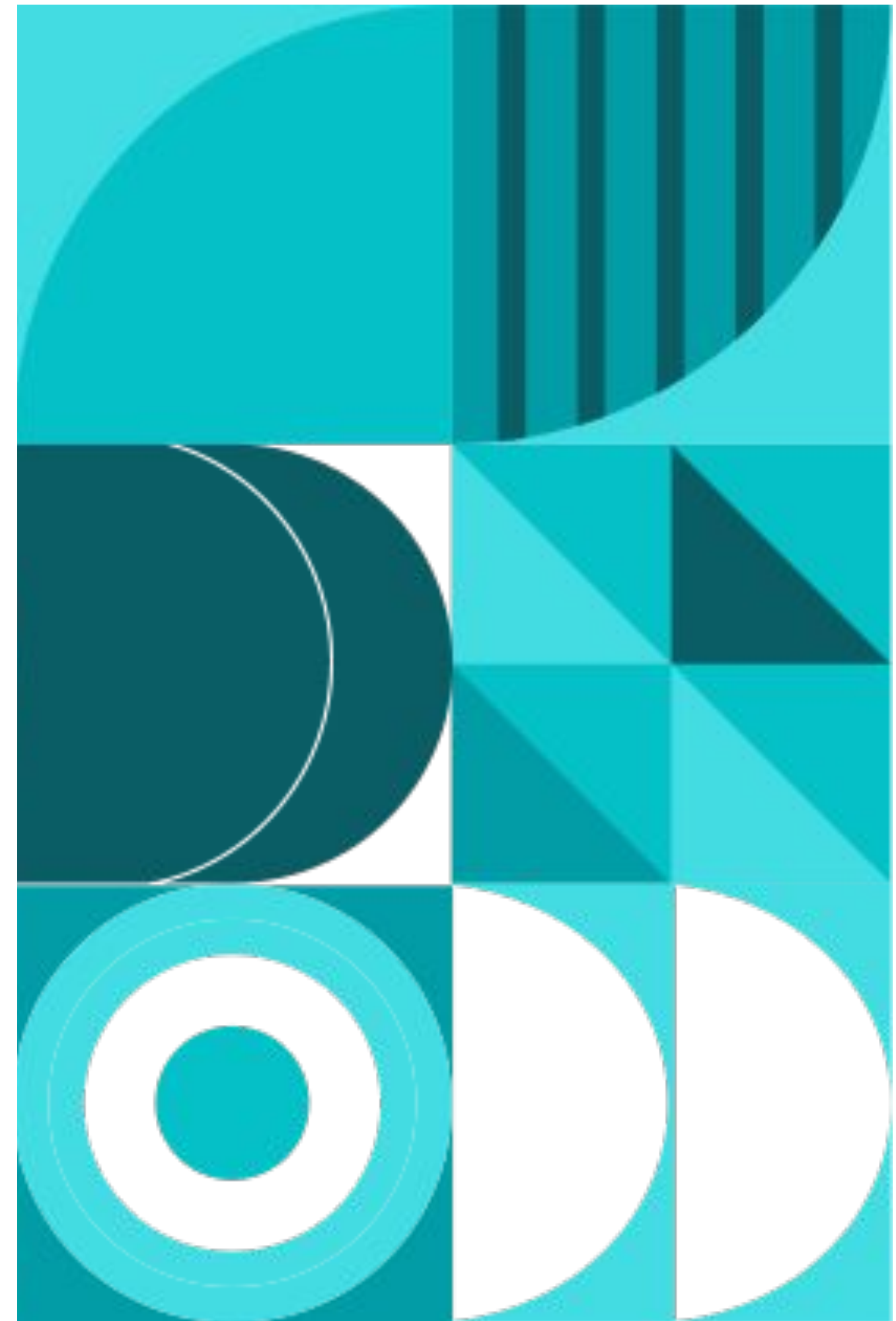


# INSERTION SORT–

## (VISUALIZE)

Insertion Sort merupakan algoritma yang bekerja dengan cara membangun larik (array) yang diurutkan satu per satu. Ia mengambil setiap elemen dari larik yang belum disatukan dan memasukkannya ke posisi yang benar dalam larik yang sudah diurutkan.

Algoritma Insertion Sort memiliki kompleksitas waktu rata-rata kasus terburuk  $O(n^2)$ . Namun kinerjanya jauh lebih baik dengan kompleksitas waktu mendekati  $O(n)$



First Pass    23 1 10 5 2 → 23 1 10 5 2

Second Pass    23 1 10 5 2 → 1 23 10 5 2

Third Pass    1 23 10 5 2 → 1 10 23 5 2

Fourth Pass    1 10 23 5 2 → 1 5 10 23 2

Fifth Pass    1 5 10 23 2 → 1 2 5 10 23



# MERGE SORT



# MERGE SORT– (VISUALIZE)

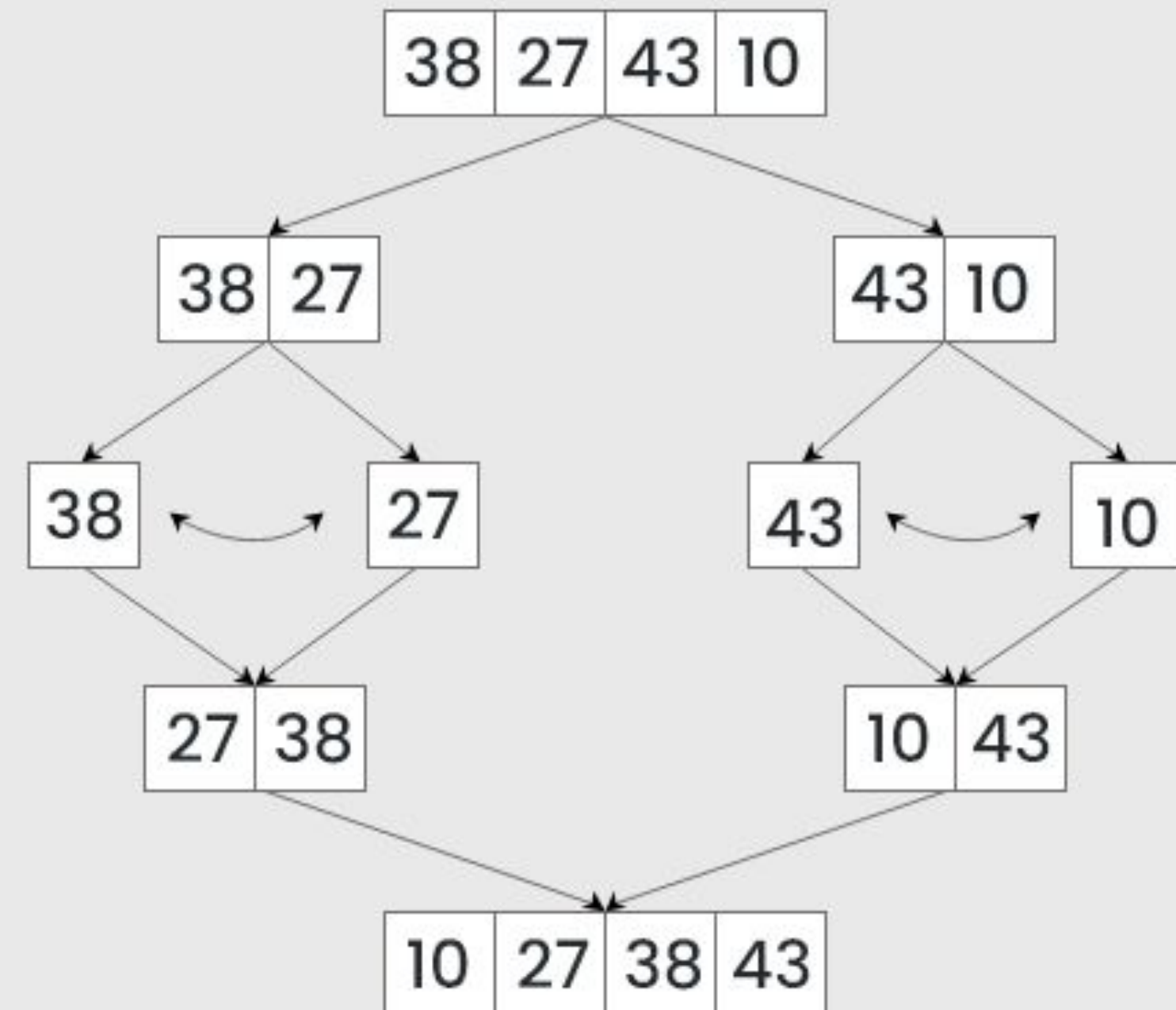
**Merge Sort** adalah algoritma yang menggunakan pendekatan “divide dan conquer” (Pecah dan Gabungkan). Algoritma ini bekerja dengan membagi larik (array) secara rekursif menjadi dua bagian hingga setiap sub-larik hanya berisi satu element. Kemudian, sub-larik tersebut digabungkan kembali secara berurutan untuk menghasilkan larik akhir yang terurut.

Algoritma Merge Sorting memiliki kompleksitas waktu  $O(n \log n)$  untuk semua kasus sehingga jauh lebih efisien dari pada insertion sort untuk kumpulan data yang besar.



# Merge Sort

Algorithm

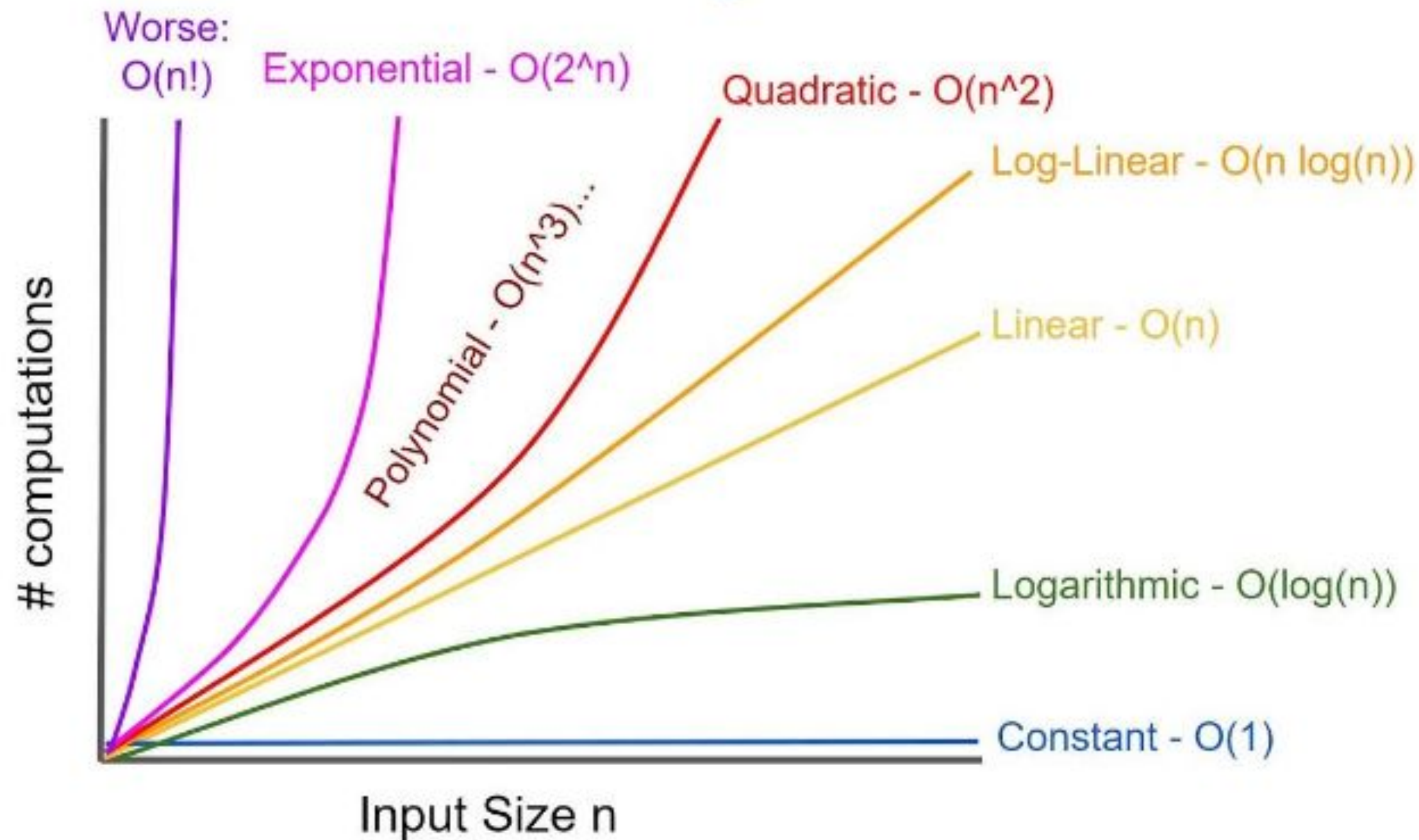




# Sorting Time Complexity

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	$O(n \log(n))$	$O(n \log(n))$	$O(n^2)$	$O(\log(n))$
Mergesort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(n)$
Timsort	$O(n)$	$O(n \log(n))$	$O(n \log(n))$	$O(n)$
Heapsort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	$O(1)$
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Shell Sort	$O(n)$	$O((n \log(n))^2)$	$O((n \log(n))^2)$	$O(1)$
Bucket Sort	$O(n+k)$	$O(n+k)$	$O(n^2)$	$O(n)$
Radix Sort	$O(nk)$	$O(nk)$	$O(nk)$	$O(n+k)$

# Sorting Time Complexity





---

# THANK YOU

---