

★ What are  
Sorting Algorithms?

- Methods/techniques used to rearrange data in a specific order (ascending or descending).
- They help organize data so it becomes easier to search, analyze, and process.

# ★ Why Do We Need Sorting Algorithms?

- Faster searching  
(binary search works only on sorted data).
- Better data organization.
- Improves performance in applications like databases, search engines, and file systems.
- Makes results easy to read and understand.



# Bubble Sort

 **Concept**

- Compare each pair of adjacent elements and swap if in wrong order.
- Like bubbles rising to the top, biggest elements keep moving to the end.

 **Real-Life Example**

- Sorting students by height in a queue:
- Two students keep comparing heights and swap until tallest reaches the end.



# Selection Sort



## Concept

- Find the minimum element from the unsorted part and put it in the correct position.
- Repeat for the next position.



## Real-Life Example

- Finding the smallest t-shirt size from a pile
- You look through the whole pile, find the smallest size, place it on top → repeat.



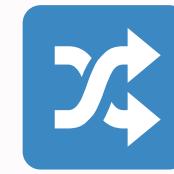
# Insertion Sort

## 👉 Concept

- Pick an element and insert it into its correct position in the already sorted part.

## 👉 Real-Life Example

- Arranging playing cards in your hand:
- You pick a new card and insert it into the right place among already sorted cards.



# Merge Sort

 **Concept**

- Divide the array into halves until single elements remain.
- Then merge them back in sorted order.
- Follows Divide & Conquer.

 **Real-Life Example**

- Dividing clothes into small piles, then merging sorted piles:
- First make small piles → sort each → merge two piles at a time.



# ⚡ Quick Sort

 **Concept**

- Pick a pivot.
- Rearrange elements: smaller on left, larger on right.
- Recursively sort left and right parts.

 **Real-Life Example**

- Partitioning guests at a dinner by age around one person (pivot):
- People younger on one side, older on the other  
→ then split and sort each side.