**Bubble Sort**

# **Introduction:**

Bubble Sort is a simple comparison-based sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The process continues until the list is sorted.

This algorithm gets its name because larger elements "bubble up" to the top (end of the array) with each pass, while smaller elements settle down at the beginning.

# **Working:**

The algorithm follows these steps:

* Start from the first element and compare it with the next one.
* If the first element is greater than the second, swap them.
* Move to the next pair and repeat the process for the entire list.
* At the end of the first pass, the largest element reaches the last position.
* Repeat the same process for the remaining elements, reducing the range in each pass.
* Continue until no swaps are needed, meaning the array is sorted.

# **Example(Dry Run):**

#### Given Array:

[5, 3, 8, 4, 2]

#### Pass 1: (Comparing and swapping adjacent elements)

* Compare 5 and 3 → Swap → [3, 5, 8, 4, 2]
* Compare 5 and 8 → No swap → [3, 5, 8, 4, 2]
* Compare 8 and 4 → Swap → [3, 5, 4, 8, 2]
* Compare 8 and 2 → Swap → [3, 5, 4, 2, 8]

#### Pass 2:

* Compare 3 and 5 → No swap → [3, 5, 4, 2, 8]
* Compare 5 and 4 → Swap → [3, 4, 5, 2, 8]
* Compare 5 and 2 → Swap → [3, 4, 2, 5, 8]

#### Pass 3:

* Compare 3 and 4 → No swap → [3, 4, 2, 5, 8]
* Compare 4 and 2 → Swap → [3, 2, 4, 5, 8]

#### Pass 4:

* Compare 3 and 2 → Swap → [2, 3, 4, 5, 8]

Since no swaps are needed after this, the array is sorted!

# **Time Complexity:**

* The best case time complexity is O(n) in case of already sorted array
* The average case and worst case time complexity will be O(n2)

# **Space Complexity:**

* The space complexity will be O(1) because bubble sort does not use any extra space thus called in place sorting algorithm

# **Optimized Bubble Sort (Using a Swap Flag):**

* The above implementation already includes an optimization using a swapped flag. This prevents unnecessary passes when the array is already sorted.

# **Advantages:**

* Simple and easy to understand
* Works well for small dataset
* Does not require extra space thus called in place sorting algorithm

# **Disadvantages:**

* Very slow for large datasets
* Have a quadratic time complexity in most cases
* Not widely used in practical applications

# **When to Use Bubble Sort?**

* If dataset is small
* If the array is nearly sorted so that the best case time complexity is achieved
* For educational purposes to understand the basics of sorting

# **Real Life Applications:**

* Teaching basic sorting concepts
* If the list is small and nearly sorted then bubble sort is used because of its simplicity
* Detecting nearly sorted array because a single pass can confirm whether an array is sorted or not

# **Conclusion:**

Bubble Sort is a simple but inefficient sorting algorithm that is mainly used for learning purposes. While it is not practical for large datasets, its simplicity makes it an ideal algorithm for beginners.