Sorting – Selection Sort

1. Introduction

Selection Sort is a simple comparison-based sorting algorithm that works by **repeatedly selecting the smallest (or largest) element** from the unsorted part of the array and placing it in its correct position.

Unlike Bubble Sort, Selection Sort makes **fewer swaps**, which helps in understanding the idea of **selection and placement**.

---

2. What is Selection Sort?

In Selection Sort:

- The array is divided into two parts:
    - **Sorted part** (left side)
    - **Unsorted part** (right side)
- In each pass, the **minimum element** from the unsorted part is selected and swapped with the first unsorted element.

---

3. Why Do We Learn Selection Sort?

Selection Sort is important because:

- It is easy to understand and implement
- Helps understand sorting fundamentals
- Introduces the idea of finding minimum or maximum
- Often used in academic learning and exams

---

4. Basic Idea Behind Selection Sort

The algorithm works as follows:

- Find the smallest element in the array

- Swap it with the first element
- Repeat the process for the remaining elements
- Continue until the array is sorted

5. Logic for Selection Sort (Plain English)

1. Start from the first element of the array
2. Assume the current position holds the minimum value
3. Compare this element with the remaining unsorted elements
4. Update the position if a smaller element is found
5. After one full pass, swap the minimum element with the first unsorted element
6. Move the boundary of the sorted part one step to the right
7. Repeat until all elements are sorted

6. Visualization of Selection Sort

Consider the array:

```
[64, 25, 12, 22, 11]
```

Pass 1:

```
Minimum = 11
Swap → [11, 25, 12, 22, 64]
```

Pass 2:

```
Minimum = 12
Swap → [11, 12, 25, 22, 64]
```

Pass 3:

```
Minimum = 22
Swap → [11, 12, 22, 25, 64]
```

Final sorted array:

```
[11, 12, 22, 25, 64]
```

7. Time and Space Complexity

Case	Time Complexity
Best Case	$O(n^2)$
Average Case	$O(n^2)$
Worst Case	$O(n^2)$

- **Space Complexity:** $O(1)$

  (In-place sorting algorithm)

8. Advantages of Selection Sort

- Simple and easy to understand
- Performs minimal number of swaps
- In-place sorting
- Works well for small datasets

9. Limitations of Selection Sort

- Inefficient for large datasets
- Time complexity remains $O(n^2)$
- Not stable by default
- Slower compared to advanced sorting algorithms

10. Real-World Applications

- Teaching sorting concepts
- Situations where swap cost is high
- Small datasets
- Academic demonstrations

11. Comparison with Bubble Sort

Feature	Bubble Sort	Selection Sort
Swaps	Many	Few
Time Complexity	$O(n^2)$	$O(n^2)$
Stability	Stable	Not stable
Efficiency	Lower	Slightly better

12. Summary

- Selection Sort selects the minimum element each pass
- Divides array into sorted and unsorted parts
- Performs fewer swaps
- Time complexity is $O(n^2)$
- Mainly used for learning purposes