Arrays –

1. Introduction

Finding the **minimum (smallest)** and **maximum (largest)** elements in an array is a **basic and frequently used operation** in data structures.

This operation helps in analyzing data, making decisions, and forming the base for more advanced algorithms like sorting and optimization problems.

2. What Do Minimum and Maximum Mean?

- **Minimum element:** The smallest value present in the array
- **Maximum element:** The largest value present in the array

Example:

```
Array: [12, 45, 7, 89, 23]
Minimum = 7
Maximum = 89
```

3. Why Do We Find Min & Max?

Finding minimum and maximum values is important because:

- It helps summarize data quickly
- Used in grading systems, analytics, and statistics
- Required in many algorithms (sorting, greedy methods)
- Helps validate data ranges

4. Basic Approach to Find Min & Max

The most common approach is to use **array traversal**.

Logical Idea:

1. Assume the first element as both minimum and maximum
2. Compare each element with current minimum
3. Compare each element with current maximum
4. Update values when required
5. Continue until the last element

This ensures every element is checked exactly once.

5. Step-by-Step Logic (Plain English)

1. Start with the first element
2. Set it as both minimum and maximum
3. Move to the next element
4. If the current element is smaller than minimum, update minimum
5. If the current element is larger than maximum, update maximum
6. Repeat until the array ends

6. Visualization of Min & Max Finding

Consider the array:

```
[10, 5, 20, 8, 30]
```

Traversal comparison:

- Start → min = 10, max = 10
- Compare 5 → min = 5
- Compare 20 → max = 20
- Compare 8 → no change
- Compare 30 → max = 30

Final result:

```
Minimum = 5
Maximum = 30
```

7. Time and Space Complexity

- **Time Complexity:** O(n)

  (Each element is visited once)

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

  (Only two extra variables are used)

This makes the approach efficient and optimal.

8. Special Cases

- **Single element array:**

  That element is both minimum and maximum

- **All elements same:**

  Minimum and maximum will be equal

- **Negative numbers:**

  Works the same way as positive numbers

9. Advantages of This Approach

- Simple and easy to understand
- No extra memory required
- Works for all types of numeric data
- Efficient for small and large arrays

10. Limitations

- Requires full traversal of the array

- Cannot be optimized further without additional logic
- Not suitable if array size is zero (empty array)

11. Real-World Applications

- Finding highest and lowest marks in a class
- Determining maximum temperature of a day
- Finding minimum price in a product list
- Analyzing sensor data ranges
- Game score analysis

12. Summary

- Finding min and max is a basic array operation
- It uses traversal and comparison
- First element is assumed initially
- Time complexity is O(n)
- Forms the base for many advanced algorithms