Arrays – Prefix Sum Technique

1. Introduction

The **Prefix Sum Technique** is an efficient method used to **precompute cumulative sums** of array elements so that **range sum queries** can be answered quickly.

Instead of repeatedly calculating sums for different ranges, prefix sums allow us to calculate them in **constant time after preprocessing**.

2. What is a Prefix Sum?

A **prefix sum array** is a new array where each element at index i stores the **sum of all elements from index 0 to index i** of the original array.

Example:

```
Original Array: [2, 4, 6, 8, 10]
Prefix Sum:     [2, 6, 12, 20, 30]
```

3. Why Do We Use Prefix Sum?

Prefix sum is used to:

- Efficiently calculate subarray sums
- Reduce time complexity for repeated range queries
- Improve performance in large datasets
- Form the base for advanced algorithms

Without prefix sum:

- Each range sum takes O(n) time

With prefix sum:

- Each range sum takes O(1) time

4. How Prefix Sum Works

Step-by-Step Logic:

1. Create a prefix sum array of the same size
2. Copy the first element as it is
3. Add the current element with the previous prefix sum
4. Repeat for all elements

5. Prefix Sum Formula

For an array arr and prefix sum array prefix:

- prefix[0] = arr[0]
- prefix[i] = prefix[i - 1] + arr[i]

To find sum of elements from index L to R:

- If L == 0:

  Sum = prefix[R]
- Else:

  Sum = prefix[R] - prefix[L - 1]

6. Visualization Example

```
Array:       [3, 1, 4, 2, 5]
Prefix Sum:  [3, 4, 8, 10, 15]
```

Find sum from index 1 to 3:

```
Sum = prefix[3] - prefix[0]
    = 10 - 3
    = 7
```

7. Time and Space Complexity

Operation	Complexity
Prefix sum construction	O(n)
Range sum query	O(1)
Space used	O(n)

8. Advantages of Prefix Sum Technique

- Extremely fast range queries
- Simple implementation
- Reduces repeated calculations
- Suitable for large input sizes

9. Limitations

- Extra memory required for prefix array
- Not suitable for frequent updates
- Requires preprocessing step
- Works best for static arrays

10. Real-World Applications

- Calculating sum of sales in a date range
- Finding total marks between roll numbers
- Analyzing traffic data over time
- Gaming score ranges
- Financial data analysis

11. Common Problems Using Prefix Sum

- Subarray sum queries
- Range sum queries
- Find equilibrium index
- Maximum subarray sum (variation)
- Difference array problems

12. Summary

- Prefix sum stores cumulative sums
- It optimizes repeated sum calculations
- Preprocessing takes O(n) time
- Range queries take O(1) time
- Widely used in competitive programming and DSA