COMPREHENSIVE OVERVIEW AND PERFORMANCE ANALYSIS

Presenter : Tran Toan

Agenda

- Introduction to Data Structures
- Memory Stack Operations
- FIFO Queue Implementation
- Sorting Algorithms Comparison
- Network Shortest Path Algorithms
- Conclusion

Introduction to Data Structures

- Definition: Data structures are ways to store and organize data to facilitate access and modifications.
- Importance: They are fundamental to efficient algorithm design and implementation.

Identify the Data Structures

Examples:

- Arrays: Fixed size, contiguous memory locations.
- Linked Lists: Elements linked using pointers.
- Stacks: LIFO structure.
- Queues: FIFO structure.
- Trees: Hierarchical structure.
- Graphs: Nodes connected by edges.
- Purpose: Each structure serves different data access and modification needs.

Define the Operations

Basic Operations:

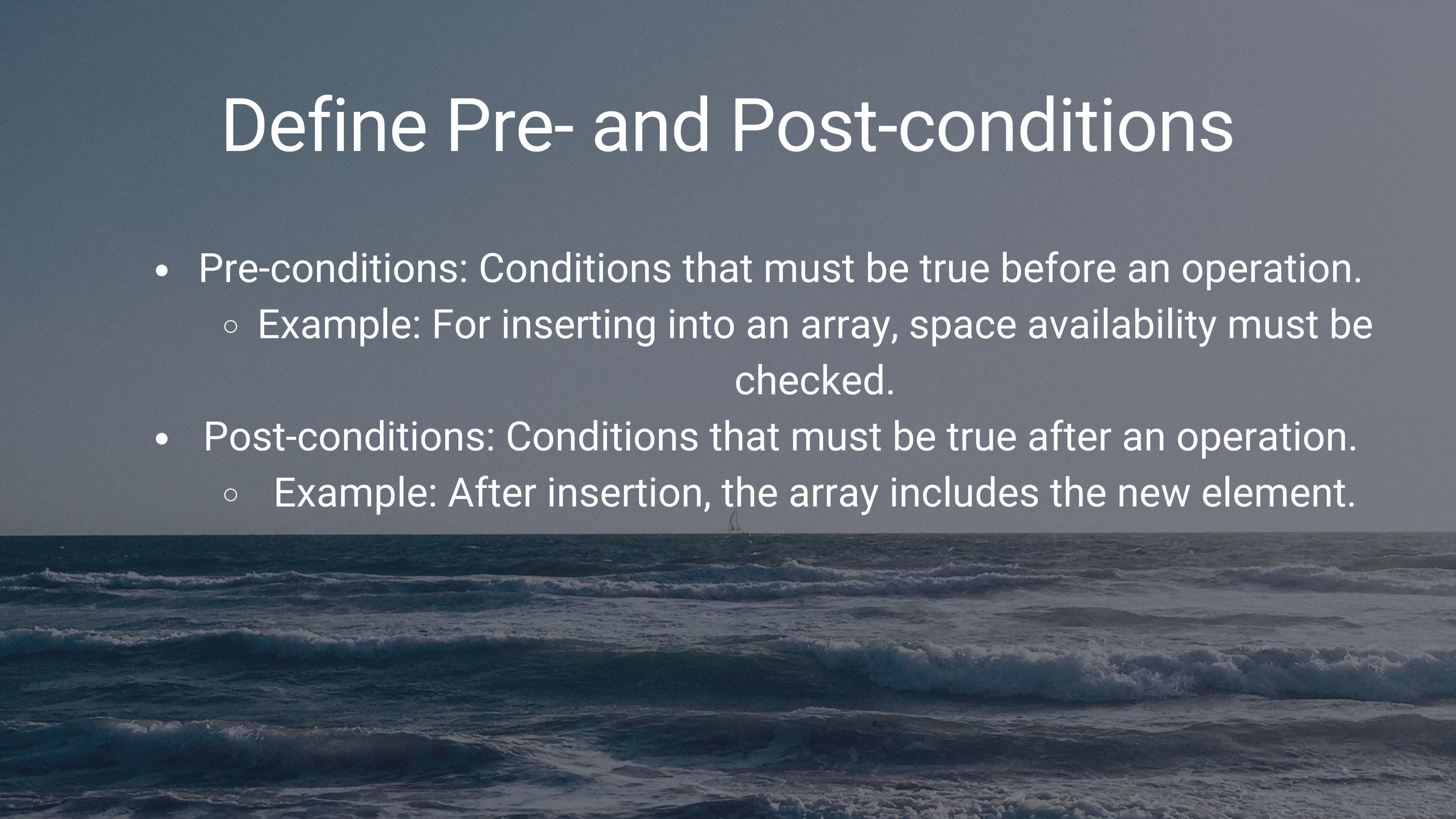
- Insertion: Adding elements.
- Deletion: Removing elements.
- Traversal: Visiting all elements.
- Searching: Finding specific elements.
- Sorting: Arranging elements in a specific order.
- Structure-Specific Operations:
- Stack: Push, Pop.
- Queue: Enqueue, Dequeue.

Specify Input Parameters

- General Parameters:
- Data values: The actual data to be stored.
- Indices: Positions within the structure.
- Pointers: References to other elements.
- Example:
- Insert operation: Requires data value and position.

Define Pre- and Post-conditions

- Pre-conditions: Conditions that must be true before an operation.
  - Example: For inserting into an array, space availability must be checked.
- Post-conditions: Conditions that must be true after an operation.
  - Example: After insertion, the array includes the new element.



Time and Space Complexity

Time Complexity:
Measure of the time an algorithm takes to complete.
Example: O(n), O(log n), O(n^2).
Space Complexity:
Measure of the memory an algorithm uses during execution.
Example: O(1), O(n).
Importance: Helps in selecting the most efficient algorithm for a task.

Define the Operations

- Example Structure: Array.
- Code Snippet: Array insertion operation in Java.
- Example :

```
public void insert(int[] arr, int value, int index) {
    if (index >= 0 && index < arr.length) {
        for (int i = arr.length - 1; i > index; i--) {
            arr[i] = arr[i - 1];
        }
        arr[index] = value;
    }
}
```

Define a Memory Stack

DEFINITION

A stack is an abstract data type that serves as a collection of elements with two principal operations:

- Push: Adds an element to the top.
- Pop: Removes the top element.

USAGE

Implements LIFO (Last In, First Out) principle.

Identify Operations

- Basic Operations:
  - Push: Add an element to the stack.
  - Pop: Remove the top element from the stack.
  - Peek: Retrieve the top element without removing it.
  - IsEmpty: Check if the stack is empty.
- Additional Operations:
  - Size: Get the number of elements in the stack.
  - Clear: Remove all elements from the stack.

Function Call Implementation

- **Explanation: Stacks are used to manage function calls in a computer.**

Mechanism:

Call Stack: Maintains the order of function calls and local variables.

Each function call creates a stack frame.

Demonstrate Stack Frames

DEFINITION

A stack frame contains a function's return address, arguments, and local variables.

VISUALIZATION

Diagram showing stack frames during nested function calls.

Discuss the Importance

- Significance:
  - Ensures proper function execution order and memory management.
- Applications:
  - Recursion: Function calls itself.
  - Backtracking: Solving problems by trying partial solutions.
  - Expression Evaluation: Evaluating mathematical expressions.

Introduction FIFO

1

Definition: A queue where the first element added is the first to be removed.

2

Acronym: First In First Out.

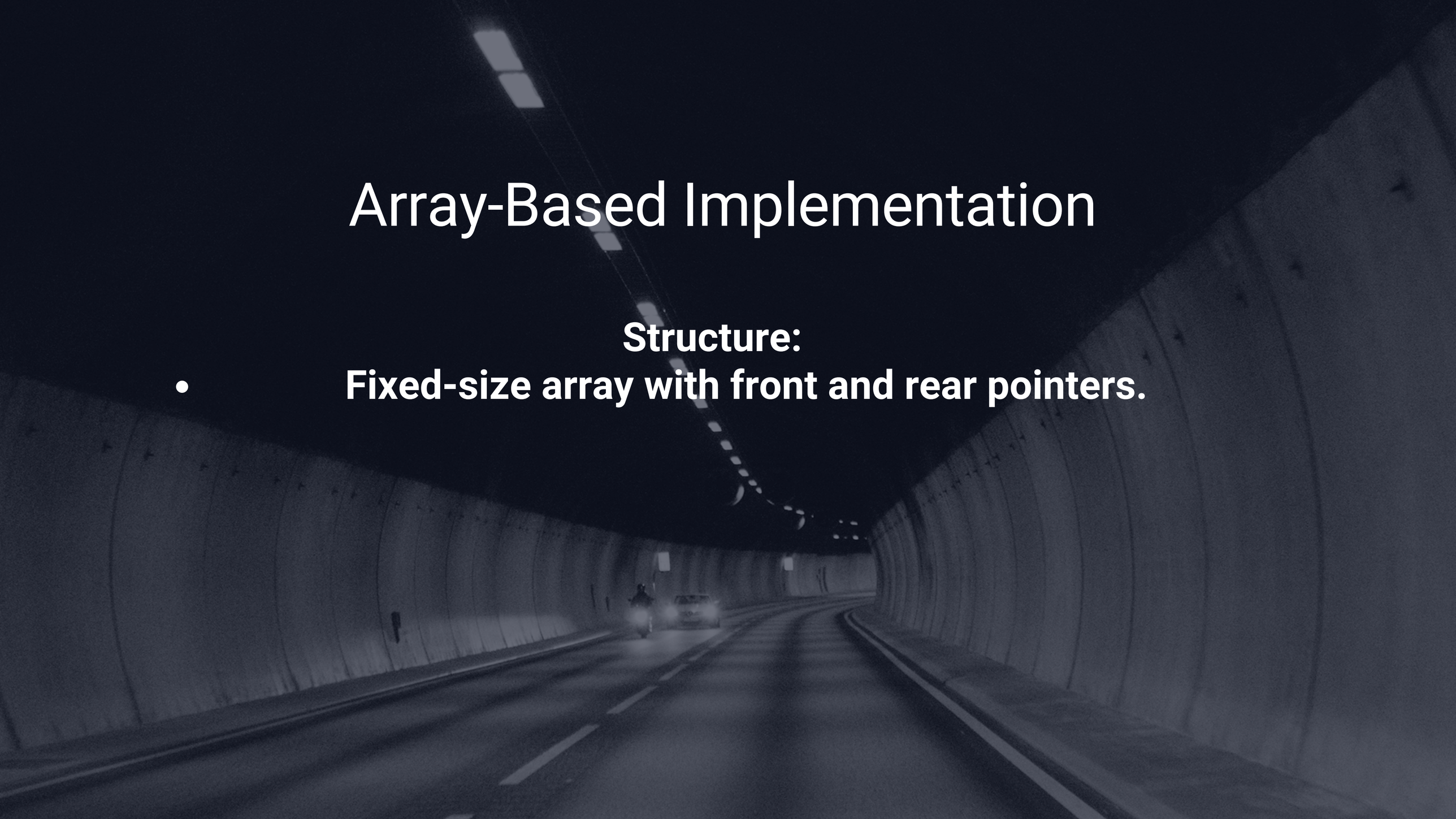
Define the Structure

Basic Structure:

- Front and Rear: Pointers to the first and last elements.
- Storage: Array or linked list.

Operations:

- Enqueue: Add an element to the rear.
- Dequeue: Remove an element from the front.
- IsEmpty: Check if the queue is empty.
- IsFull: Check if the queue is full (for fixed-size array implementation).



Array-Based Implementation

Structure:

- **Fixed-size array with front and rear pointers.**

Linked List-Based Implementation

Structure:

- Dynamic size with nodes pointing to the next element.

FIFO Queue Example

- Scenario: Print job management.
- Illustration: Diagram showing enqueue and dequeue operations.

Introduction to Sorting Algorithms

1

Overview: Importance of sorting for data organization and retrieval.

2

Algorithms: Bubble Sort and Quick Sort

Time Complexity Analysis

BUBBLE SORT

O(n^2) in the worst case.

QUICK SORT:

O(n log n) on average.

COMPARISON

Bubble Sort is generally slower than Quick Sort.

Space Complexity Analysis

BUBBLE SORT

O(1) extra space.

QUICK SORT:

O(log n) extra space for the recursive stack.

COMPARISON

Quick Sort uses more space due to recursion.

Stability

- Definition: Stable sorting maintains the relative order of equal elements.
- Bubble Sort: Stable.
- Quick Sort: Not stable.
- Importance: Stability is crucial for certain applications where the relative order of equal elements matters.

Comparison Table

Criteria	Bubble Sort	Quick Sort
Time Complexity	O(n^2)	O(n log n)
Space Complexity	O(1)	O(log n)
Stability	Stable	Not stable
Performance	Slow	Fast

Performance Comparison

- Analysis: Empirical performance on different data sets.
- Example: Execution time comparison on a large dataset.
  - Bubble Sort: Slow for large datasets.
  - Quick Sort: Efficient for large datasets.

Introduction to Network Shortest Path Algorithms

Overview: Finding the shortest paths in weighted graphs

Algorithms: Dijkstra's Algorithm and Prim-Jarnik Algorithm.

Dijkstra's Algorithm

- Explanation: Greedy algorithm for finding the shortest path from a single source to all other nodes in a weighted graph.
- Steps:

1. Initialize distances from the source to all nodes as infinity.
2. Set distance to the source itself as zero.
3. Use a priority queue to explore nodes with the shortest known distance.
4. Update the distance to neighboring nodes.

- Example: Step-by-step illustration on a sample graph.

Prim-Jarnik Algorithm

- Explanation: Algorithm for finding the minimum spanning tree of a weighted undirected graph.
- Steps:
  a. Start with a single vertex.
  b. Grow the minimum spanning tree by adding the smallest edge connecting the tree to a vertex not yet in the tree.
  c. Repeat until all vertices are included.
- Example: Step-by-step illustration on a sample graph.

Thanks and see you again !!!!