Queue

Introduction to STL Queue:

- The Standard Template Library (STL) Queue is a container adapter in C++ that provides a FIFO (First-In-First-Out) data structure.
- It is part of the C++ Standard Library and implemented as an adapter over other container classes (e.g., deque or list) to provide queue-like functionality.

Use Cases: STL Queue is commonly used in various scenarios:

BFS (Breadth-First Search) Algorithm:

- To traverse and explore nodes in a graph level-by-level.

Implementing Caching Mechanism:

- To store temporary data in the cache with a limited capacity, removing the oldest entry when the cache is full.

Print Queue-like Outputs:

- To manage a printing queue, where new print jobs are added to the back, and the printer serves them from the front.

Task Scheduling:

- To implement a task scheduler, where tasks are added to the queue and executed in the order they were added.

Limitations of STL Queue:

- The STL Queue does not provide direct iterators for traversal like other containers (e.g., vector, list).
- It does not support random access to elements, as it follows a strict FIFO order.

The Time and Space complexity of the functions used:

- **push()**: Inserts an element to the back of the queue.
    - Time complexity: O(1)
    - Space complexity: O(1)
- **empty()**: Checks if the queue is empty.
    - Time complexity: O(1)
    - Space complexity: O(1)
- **size()**: Returns the number of elements in the queue.

    - Time complexity: O(1)
    - Space complexity: O(1)
- **front()**: Accesses the front element of the queue.
    - Time complexity: O(1)
    - Space complexity: O(1)
- **back()**: Accesses the back element of the queue.
    - Time complexity: O(1)
    - Space complexity: O(1)
- **pop()**: Removes the front element from the queue.
    - Time complexity: O(1)
    - Space complexity: O(1)