

Stack and Queue Operations

Stack Operations:

1. **Push:** Adds an element to the top of the stack.
2. **Pop:** Removes and returns the element at the top of the stack.
3. **Peek/Top:** Returns the element at the top of the stack without removing it.
4. **isEmpty:** Checks if the stack is empty or not.
5. **Size:** Returns the number of elements in the stack.

Queue Operations:

1. **Enqueue:** Adds an element to the rear of the queue.
2. **Dequeue:** Removes and returns the element at the front of the queue.
3. **Front:** Returns the element at the front of the queue without removing it.
4. **isEmpty:** Checks if the queue is empty or not.
5. **Size:** Returns the number of elements in the queue.

Key Differences:

1. **Ordering:** Stack follows Last In First Out (LIFO) ordering while Queue follows First In First Out (FIFO) ordering.
2. **Implementation:** Stack can be implemented using arrays or linked lists. Queue can also be implemented using arrays or linked lists.
3. **Applications:** Stacks are used in function calls, undo mechanisms, and expression evaluation. Queues are used in scheduling, buffering, and breadth-first search algorithms.

Understanding these basic operations of stacks and queues is essential for implementing various data structures and algorithms efficiently.