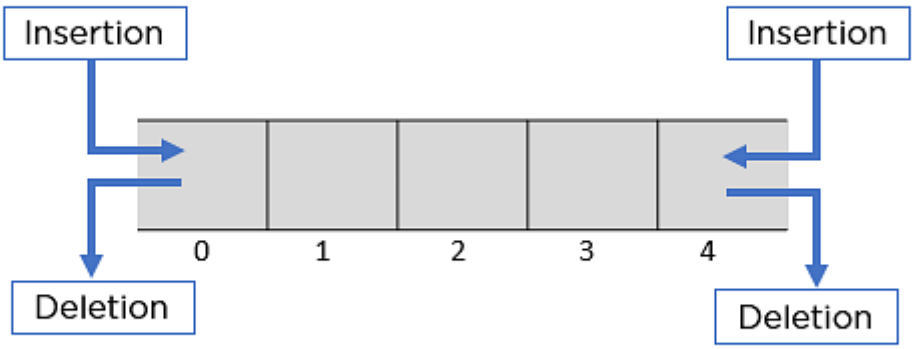
Deque in Data Structure

Deque is a data structure that inherits the properties of both queues and stacks. Additionally, the implementation of this data structure requires constant time, i.e., time complexity = O(1). This means you can use deque to your advantage to implement both the queue and stack.

What is Deque in Data Structure?

Deque is a more generalized version of a linear queue. As you know, the linear queue has some restrictions while performing the insertion and deletion of elements. The insertion in a linear queue must happen from the rear end and deletion from the front end. But, in deque, you can perform both insertion and deletion operations at both of its ends. That's why it is called a Double-Ended Queue (Deque).

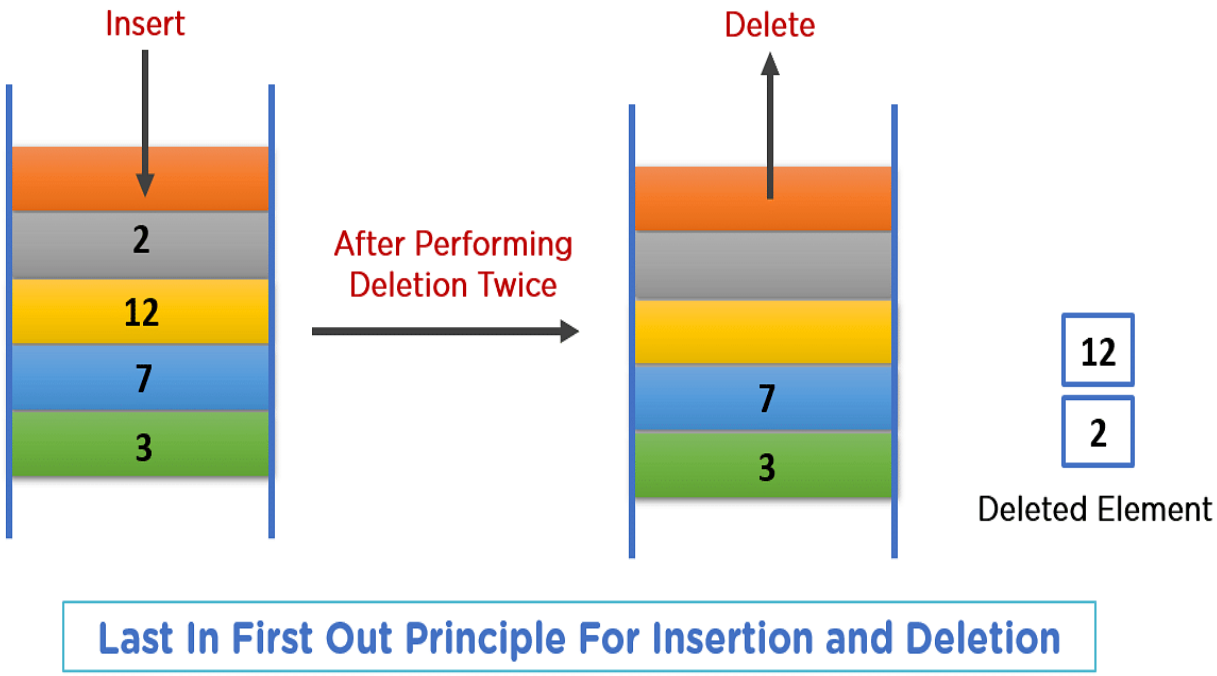


Properties of Deque

Now, you will look into the primary properties of deque in a data structure.

- A deque can perform both the insertion and deletion using the LIFO (Last In First Out) principle.

The insertion and deletion in deque can be limited to one node. When you do that, the deque becomes a stack. Hence, it follows the Last In First Out Principle for insertion and deletion.

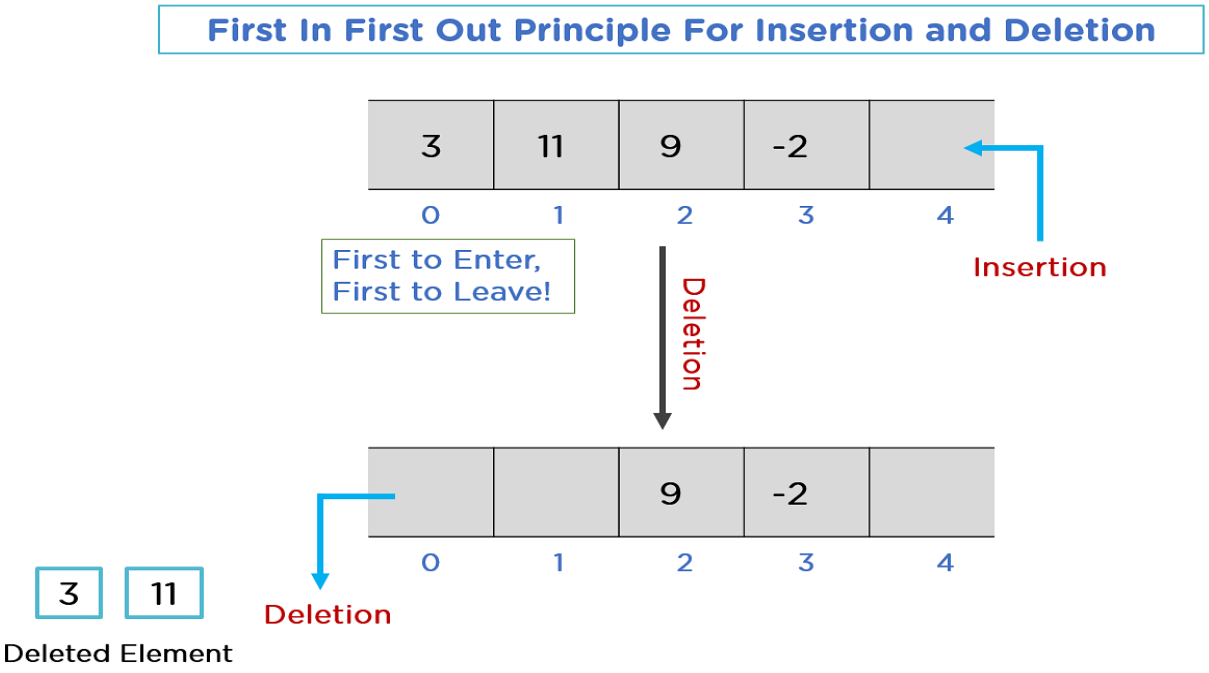


Last In First Out Principle For Insertion and Deletion

As shown in the image above, initially, there were four elements inside this deque. However, after repeating the removal of an element twice, data elements 2 and 12 get removed from the deque, respectively. You inserted element 2 at last and removed it at first. This example demonstrates the LIFO principle.

- A Deque can perform insertion and deletion operations using the FIFO (First In First Out) principle.

As deque can operate on both ends, you can extract queue properties by limiting insertion at the front node and removal at the rear node. Doing this will allow deque to act as a linear queue.



As shown in the image above, the element which enters at first will also leave the queue first. This is how deque can use the FIFO principle to perform insertion and deletion.