

Stack & Queue

"IMPLEMENT STACK USING DEQUE"

Before learning how to implement stack using deque, let us understand what is stack & deque.

Stack:

A stack is a linear data structure that follows: LIFO rules (Last In First Out). This means the element inserted last will be removed first.

Deque (double-ended queue):

A deque (pronounced as 'deck') is a double-ended queue where elements can be inserted or deleted from both the ends. (front & rear end).

Implementation of stack using deque:

We can implement a stack using deque (double-ended queue) from Python's collections module, as it provides efficient $O(1)$ time complexity for both insertion & deletion at the end.

Steps to implement a stack using deque.

Step 1: Initialize the stack

- create a class Stack
- inside the constructor (-init) define empty deque to store stack elements.

Step 2: Push operation

- use deque.append(x) to insert an element at the right end.

Step 3: pop operation

- use deque.pop() to remove the last inserted element

Step 4: peep operation

- use deque[-1] to retrieve the top element without removing it.

Step 5: check if stack is empty

- check if the deque has zero elements (len(self.stack) == 0)

Step 6: Get stack size

- return the no of elements using len(self.stack)