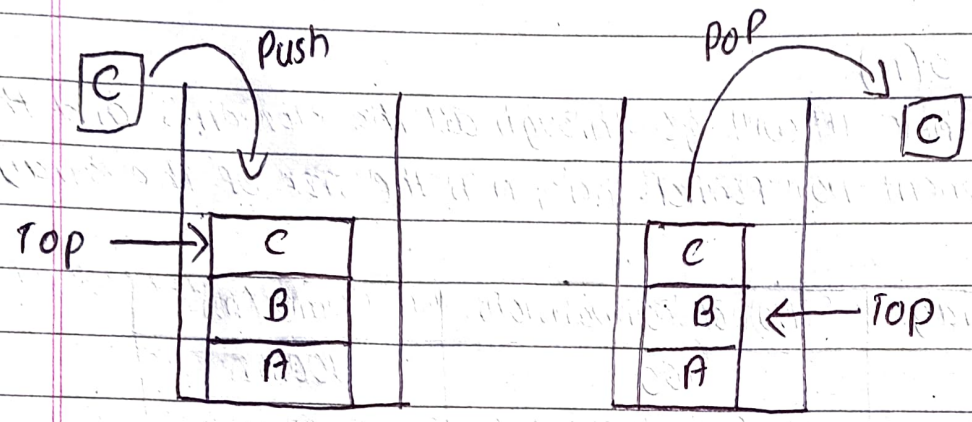
Date
Page No.

Stack Data structure

- Stack is a linear Data structure that Pollows a particular order in which the operations are perPormed.
- The order may be a LIFO (Last In First Out) or FILO (First In Last Out)



- It behaves like a stack of plates, where the last plate added is the First one to be removed. Think of it as this way.
  - o Pushing an element onto the stack is like adding a new plate on top
  - o Popping an element removes the plate Prom the stack
- Stack is also known as a push down list.
- In an Empty stack : TOP = -1

- Stack as an ADT: (Operations)
  - CreateEmptyStack (S)
  - Push (S, x)
  - Top (S)
  - Pop (S)
  - IsFull (S)
  - IsEmpty (S)

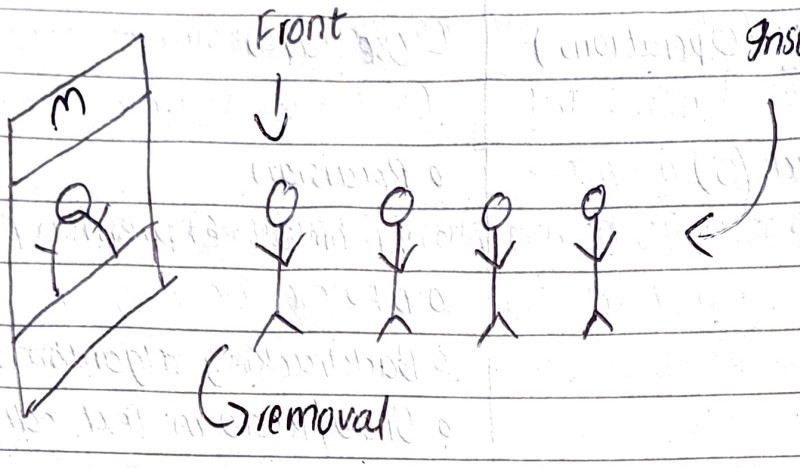
- Use cases
  - Recursion
  - Arithmetic expression parsing
  - DFS
  - Backtracking algorithms
  - Undo/Redo in text editors

- Internally it is an array & it extends vector class
- Time Complexity : O(1) (For, Access, Search, Insertion Deletion) [Best Case]
- Time Complexity : O(1) (For, Insertion and Deletion)
  O(N) (For, Access and Search) [Worst Case]
- Average Time Complexity : Same as worst case
- Space Complexity: O(n)

- Stack is a class in JAVA

Queue Data Structure

- Queue is a linear data structure that follows a particular order in which the operations are performed.
- The order it follows is FIFO (First In First Out)



- It behaves as a line at a bank or at a bus stop or a waiting list of students are real life examples of a queue.
  - Adding a person at the end of the line is known as enqueue.
  - Removing a person from the front of the line is known as dequeue.

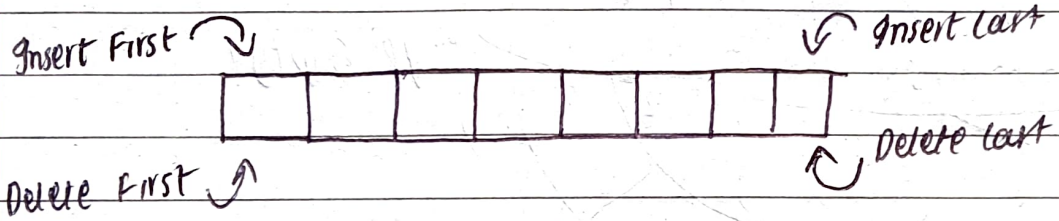
- Queue as an ADT: (operations)	- Use cases	
o MakeEmpty(q)	o CPU scheduling	OS
o Enqueue(q, x)	o I/O Buffers	OS
o Dequeue(q)	o Job Scheduling	OS
o IsFull(q):	o Packet Queuing	Networking
o IsEmpty(q)	o Message Queues	Networking
o Traverse(q):	o BFS	
o peek(q):	o Task Scheduling	
	o Elevator Systems	
	o Printer Queue	

- Internally it is a linear array & it is an interface in JAVA

- Time complexity : O(1) (For, Access, search, Insertion, Deletion) (Best Case)
  Time complexity : O(N) (For: Access, search), O(1) (For: Insertion, Deletion)
  Space complexity : O(N)

Deque Data Structure

- A deque is a linear data structure that allows insertion and deletion from the front and the rear ends
- Unlike a regular queue or stack, which restrictions (restrict) operations to one end



- Supported operations
  - push-front(x)
  - push-back(x)
  - pop-front(x)
  - pop-back(x)
  - peek-front(x)
  - peek-back(x)
  - is_empty(x)
  - size()

- Use Cases
  - Browser history navigation
  - Task scheduling systems
  - Sliding window algorithms
  - Palindrome checking

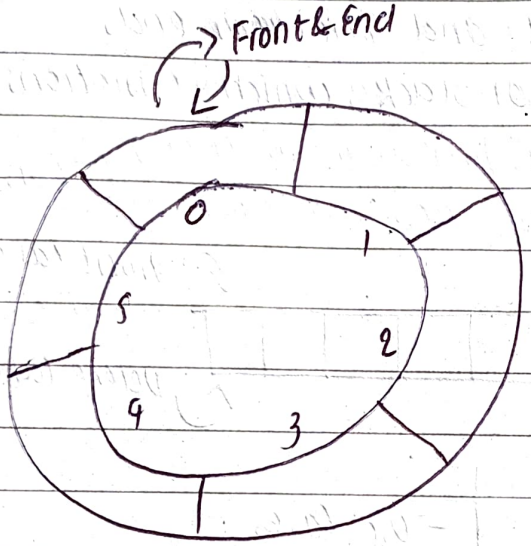
In an empty Queue Top = -1

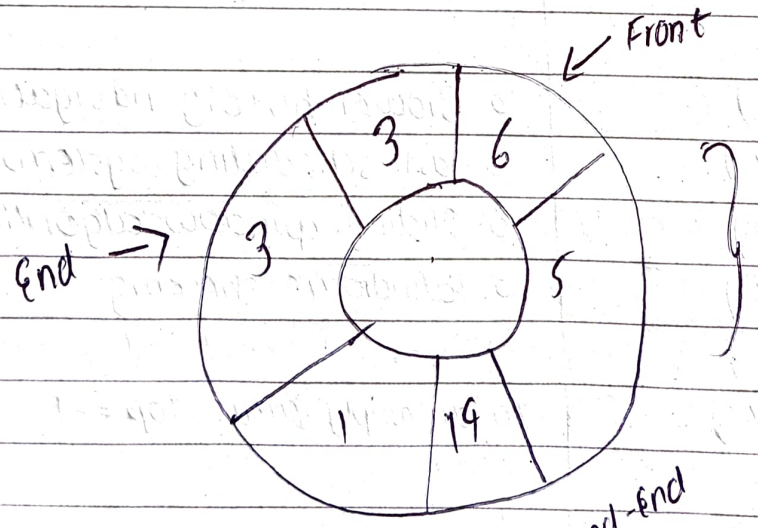
- Space complexity: O(N)
- Time complexity: O(1) [Best & Worst case] O(N)

- Deque is a more flexible verson of a queeue.

Circular queue

- Circular Queue is a dinear data structure that connects the end of the queeue back to the Piont to make use of efficient space.





- Operations : (Similar to normal Queue)
- Time Complexity : O(1)
- Space Complexity : O(N)

- How it works :

  - rear = (rear + 1) % size -> Circular increment
  - front = (front + 1) % size -> Circular increment
  - (rear + 1) % size == front (if full)
  - front == -1 (if empty)

- Efficient space usage.