

# CE205 Data Structures

## Week-3

Stacks, Queue Structures, and Related Algorithms and Problems.

Download [DOC](#), [SLIDE](#), [PPTX](#)

## Outline-1

- Stack ADT
  - Stack Using Array
  - Stack Using Linked List
- Expressions
  - Infix
  - Postfix
  - Prefix
  - Infix to Postfix Conversion
  - Postfix Expression Evaluation

## Outline-2

- Queue ADT
  - First Come First Serve, FCFS, FIFO
  - Queue Data structure Using Array
  - Queue Using Linked List
  - Circular Queue Data structure
  - Double Ended Queue Data structure
  - Multilevel Queue (MLQ)
- Hanoi Tower

## Stack ADT

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/stack-adt.html](http://www.btechsmartclass.com/data_structures/stack-adt.html)

## Stack Using Array

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/stack-using-array.html](http://www.btechsmartclass.com/data_structures/stack-using-array.html)

## Stack Using Linked List

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/stack-using-linked-list.html](http://www.btechsmartclass.com/data_structures/stack-using-linked-list.html)

## Expressions

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/expressions.html](http://www.btechsmartclass.com/data_structures/expressions.html)
    - Infix
    - Postfix
    - Prefix

## Infix to Postfix Conversion

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/infix-to-postfix.html](http://www.btechsmartclass.com/data_structures/infix-to-postfix.html)



## Postfix Expression Evaluation

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/postfix-evaluation.html](http://www.btechsmartclass.com/data_structures/postfix-evaluation.html)

## Queue ADT

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/queue-adt.html](http://www.btechsmartclass.com/data_structures/queue-adt.html)

## First Come First Serve, FCFS, FIFO

- BTech Smart Class
  - <http://www.btechsmartclass.com/downloads/lab-manuals/Operating-System-Lab-Manual-R18-JNTUH.pdf>

## Queue Data structure Using Array

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/queue-using-array.html](http://www.btechsmartclass.com/data_structures/queue-using-array.html)

## Queue Using Linked List

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/queue-using-linked-list.html](http://www.btechsmartclass.com/data_structures/queue-using-linked-list.html)

## Circular Queue Data structure

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/circular-queue.html](http://www.btechsmartclass.com/data_structures/circular-queue.html)

## Double Ended Queue Data structure

- BTech Smart Class
  - [http://www.btechsmartclass.com/data\\_structures/double-ended-queue.html](http://www.btechsmartclass.com/data_structures/double-ended-queue.html)

## Multilevel Queue (MLQ)

- Geeks for Geeks
  - <https://www.geeksforgeeks.org/multilevel-queue-mlq-cpu-scheduling/>



## Hanoi Tower

- Geeks for Geeks
  - Recursive Version
    - [Program for Tower of Hanoi - GeeksforGeeks](#)
  - Iterative Version
    - [Iterative Tower of Hanoi - GeeksforGeeks](#)

## Hanoi Tower Iterative Algorithm:

S = Source

A = Aux

D = Dest

Calculate the total number of moves required i.e.

$\text{pow}(2, n) - 1$  here n is number of disks.

## Hanoi Tower Iterative Algorithm:

- If number of disks (i.e.  $n$ ) is even then interchange destination pole and auxiliary pole.
- for  $i = 1$  to total number of moves:
  - if  $i \% 3 == 1$ :
    - legal movement of top disk between source pole and destination pole
  - if  $i \% 3 == 2$ :
    - legal movement top disk between source pole and auxiliary pole
  - if  $i \% 3 == 0$ :
    - legal movement top disk between auxiliary pole and destination pole

*End – Of – Week – 3*