

CE205 Data Structures Week-3

Stacks, Queue Structures and Related Algorithms and Problems.

Author: Asst. Prof. Dr. Uğur CORUH

Contents

1	CE205 Data Structures	1
1.0.1	Week-3	1
1.0.2	Outline-1	1
1.0.3	Outline-2	2
1.0.4	Stack ADT	2
1.0.5	Stack Using Array	2
1.0.6	Stack Using Linked List	2
1.0.7	Expressions	2
1.0.8	Infix to Postfix Conversion	2
1.0.9	Postfix Expression Evaluation	3
1.0.10	Queue ADT	3
1.0.11	First Come First Serve, FCFS, FIFO	3
1.0.12	Queue Data structure Using Array	3
1.0.13	Queue Using Linked List	3
1.0.14	Circular Queue Data structure	3
1.0.15	Double Ended Queue Data structure	3
1.0.16	Multilevel Queue (MLQ)	3
1.0.17	Hanoi Tower	3
1.0.18	Hanoi Tower Iterative Algorithm:	4
1.0.19	Hanoi Tower Iterative Algorithm:	4

List of Figures

List of Tables

1 CE205 Data Structures

1.0.1 Week-3

1.0.1.1 Stacks, Queue Structures, and Related Algorithms and Problems. Download DOC¹, SLIDE², PPTX³

1.0.2 Outline-1

- Stack ADT
 - Stack Using Array
 - Stack Using Linked List

¹ce205-week-3-stack.md_doc.pdf

²ce205-week-3-stack.md_slide.pdf

³ce205-week-3-stack.md_slide.pptx

- Expressions
 - Infix
 - Postfix
 - Prefix
 - Infix to Postfix Conversion
 - Postfix Expression Evaluation
-

1.0.3 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
-

1.0.4 Stack ADT

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/stack-adt.html
-

1.0.5 Stack Using Array

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/stack-using-array.html
-

1.0.6 Stack Using Linked List

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/stack-using-linked-list.html
-

1.0.7 Expressions

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/expressions.html
 - * Infix
 - * Postfix
 - * Prefix
-

1.0.8 Infix to Postfix Conversion

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/infix-to-postfix.html
-

1.0.9 Postfix Expression Evaluation

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/postfix-evaluation.html
-

1.0.10 Queue ADT

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/queue-adt.html
-

1.0.11 First Come First Serve, FCFS, FIFO

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

1.0.12 Queue Data structure Using Array

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/queue-using-array.html
-

1.0.13 Queue Using Linked List

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/queue-using-linked-list.html
-

1.0.14 Circular Queue Data structure

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/circular-queue.html
-

1.0.15 Double Ended Queue Data structure

- BTech Smart Class
 - http://www.btechsmartclass.com/data_structures/double-ended-queue.html
-

1.0.16 Multilevel Queue (MLQ)

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

1.0.17 Hanoi Tower

- Geeks for Geeks
 - Recursive Version
 - * Program for Tower of Hanoi - GeeksforGeeks⁴

⁴<https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/>

- Iterative Version
 - * Iterative Tower of Hanoi - GeeksforGeeks⁵
-

1.0.18 Hanoi Tower Iterative Algorithm:

S = Source

A = Aux

D = Dest

Calculate the total number of moves required i.e.

$pow(2, n) - 1$ here n is number of disks.

1.0.19 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

⁵<https://www.geeksforgeeks.org/iterative-tower-of-hanoi/>