Home Work

CS221: Data Structures and Algorithms

Usman Institute of Technology

Fall 2019

Stack and Queue

1. Consider a Stack of size 5 and perform following operations. Demonstrate this by creating a separate stack for each operation.
    a. Push(4)
    b. Push(3)
    c. Pop()
    d. Push(8)
    e. Pop()

2. Consider a Queue of size 5 and perform following operations. Demonstrate this by creating a separate stack for each operation.
    a. Enqueue(4)
    b. Enqueue(3)
    c. Deqeueue()
    d. Enqueue(8)
    e. Dequeue()

3. Perform Question 1 & 2 for the Stack and Queue, implemented using array, respectively. Demonstrated each operation by creating an array and depict the movement of pointer variables.

4. Perform following questions from text book:
    a. 10.1-2
    b. 10.1-5
    c. 10.1-6
    d. 10.1-7

5. Write a program to convert infix expression into post-fix expression using Stack

6. Write a program to solve an post-fix expression using Stack

7. Provide code for a Queue in which each time when you perform Dequeue(), it removes the element with highest value instead of on the basis of FIFO. However, if there are two elements with same value then it must return using FIFIO for those elements.

    For example, consider a Queue
    - Pus(5)
    - Push(7)
    - Push(3)
    - Push(5)
    - Pop() = will return 7

- Pop() = will return 5 – which inserted the first

Also identify the time complexity for each operation of this queue.

8. Implement Stack ADT with the following operations whereas time complexity of each operation for your implementation must be as given:
    a. Push O(1)
    b. Pop O(1)
    c. Peek O(1)
    d. IsEmpty O(1)
    e. IsFull O(1)
    f. Count O(1)
    g. Clear O(1)
9. Implement Stack ADT with the following operations whereas time complexity of each operation for your implementation must be as given:
    a. Enqueue O(1)
    b. Dequeue O(1)
    c. Peek O(1)
    d. IsEmpty O(1)
    e. IsFull O(1)
    f. Count O(1)
    g. Clear O(1)

10. Write a program to check whether the given expression has balanced parenthesis or not. The expression may contain digits, operators (+, -, /, *, %), and parenthesis ((,), [,], {, }). For example, the following expression has the balanced parenthesis:

    {2 + 2 * (5+3)} / [4 + 8]