Date : 1st - 10 - 2020
Morning Session : 9am – 11.00 PM
By ~ Rohan Kumar

Topics: Stack Part-2

Evaluations of Expressions:(Application Of Stack)

1) **Infix**
2) **Prefix**
3) **postfix**

Expression: a + b,
Expressions contains constans , variables, and symbols

Infix : Infix expression if the operator appears in between the operands in the expression. Simply of the form (operand1 operator operand2).
Example : (A+B) * (C-D)

Prefix : prefix expression if the operator appears in the expression before the operands. Simply of the form (operator operand1 operand2).

Postfix:postfix expression if the operator appears in the expression after the operands. Simply of the form (operand1 operand2 operator).

For Code Please go through the Lecture:

Recorded Lecture

MCQ 1:

1. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: (()(())(())) are:
   a) 1
   b) 2
   c) 3
   d) 4 or more

Answer: C, 3

MCQ 2:

2. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. Suppose that you run the algorithm on a sequence that contains 2 left parentheses and 3 right parentheses (in some order).
   The maximum number of parentheses that appear on the stack AT ANY ONE TIME during the computation?
   a) 1
   b) 2
   c) 3
   d) 4 or more

Answer: B, 2

MCQ 3:

3. What is the value of the postfix expression 6 3 2 4 + – *?
   a) 1
   b) 40
   c) 74
   d) -18

Answer: D , -18

MCQ 4 :

4. Here is an infix expression: 4 + 3*(6*3-12). Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation.
   The maximum number of symbols that will appear on the stack AT ONE TIME during the conversion of this expression?
   a) 1
   b) 2
   c) 3
   d) 4

Answer: D, 4