# Data Structures and Algorithms

**Lecture 16:** Stacks – Polish notation, Evaluation and transformation of expressions

# Polish Notation

Polish Notation in data structure tells us about different ways to write an arithmetic expression.

- An arithmetic expression contains 2 things, i.e., operands and operators.
- Operands are either numbers or variables that can be replaced by numbers to evaluate the expressions.
- Operators are symbols symbolizing the operation to be performed between operands present in the expression.
- Like the expression (1+2)∗(3+4) standard becomes ∗+12+34 in Polish Notation.

# Polish Notation

- Polish notation is also called **prefix** notation.
- It means that operations are written before the operands.
- The operators are placed left for every pair of operands.
- Let’s say for the expression a+b, the prefix notation would be +ab.

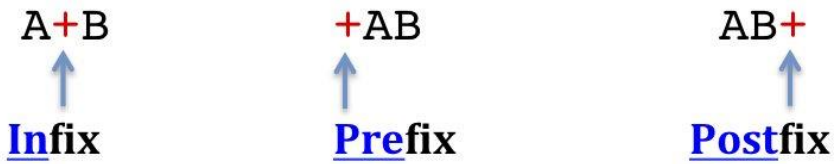
# Types of Polish Notation

Three types of polish notations exist in the data structure.

Infix Notation

i. (3+7)

ii. (1*(2+3))




Prefix Notation

i. 3+7 will convert into +37

ii. 1*(2+3) will convert into *1(+23)

Postfix Notation

i. 3+7 will convert into 37+

ii. 1*(2+3) will convert into (23+)1*

# Polish Notations

**Conversion of an Infix Expression to Postfix Expression:**

- To convert each expression from infix to postfix, we assign priority to each of the operators present in the expression.
- Each operator has its priority for an expression.
- For example, if we take some operators, i.e., +, -, *, /, then these will be arranged in priority.
  - Higher Priority Operators : *, /, %.
  - Lower Priority Operators : +, -.
  - Order of Operators : +, −, ∗, /, ^

# Polish Notations

**Conversion of an Infix Expression to Postfix Expression:**

- Push "(" in the stack and add ")" at the end of the infix polish notation of the given expression.
- Repeat the below steps for each of the elements present in the Infix Polish Notation.
- If "(" is encountered, then push the element onto the stack.
- If an operand, i.e. a variable or a number, is encountered, we add it in the postfix or reverse polish notation.

# Polish Notations

**Conversion of an Infix Expression to Postfix Expression:**

- If ")" is encountered, we pop from the stack until the popped element is "(" and add these elements to the postfix expression. After that discard "(" from the stack and do not add it again to the postfix expression.
- If an operator "x" is encountered, then, again and again, pop from the stack and add each operator to the postfix expression which has the same or higher precedence than operator "x".
- After performing step 2 on each element, we will pop all the elements from the stack and add them to the postfix expression.

# Conversion of an Infix Expression to Postfix Expression

**Expression: A+CB*C)**

| Character Scanned | Stack | Postfix Expression |
|---|---|---|
| A | C | A |
| + | C | A |
| C | C+ | A |
| B | C+C | AB |
| * | C+C | AB |
| C | C+C* | ABC |
| ) | C+C* | ABC*+ |

**Output : ABC*+**

# Conversion of Infix to Prefix using Stack

**Rules:**

- First, reverse the infix expression given in the problem.
- Scan the expression from left to right.
- Whenever the operands arrive, print them.
- If the operator arrives and the stack is found to be empty, then simply push the operator into the stack.
- If the incoming operator has higher precedence than the TOP of the stack, push the incoming operator into the stack.
- If the incoming operator has the same precedence with a TOP of the stack, push the incoming operator into the stack.

# Conversion of Infix to Prefix using Stack

**Rules:**

- If the incoming operator has lower precedence than the TOP of the stack, pop, and print the top of the stack. Test the incoming operator against the top of the stack again and pop the operator from the stack till it finds the operator of a lower precedence or same precedence.
- If the incoming operator has the same precedence with the top of the stack and the incoming operator is ^, then pop the top of the stack till the condition is true. If the condition is not true, push the ^ operator.
- When we reach the end of the expression, pop, and print all the operators from the top of the stack.

# Conversion of Infix to Prefix using Stack

**Rules:**

- If the operator is ')', then push it into the stack.
- If the operator is '(', then pop all the operators from the stack till it finds ) opening bracket in the stack.
- If the top of the stack is ')', push the operator on the stack.
- At the end, reverse the output.

# Recall:

- Polish notations and its types
- Expression conversion