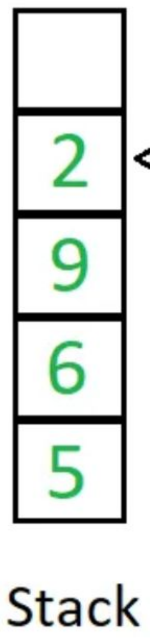
Stack

Md. Tanvir Alam

Basic Stack Operations

- Push
- Pop
- Peek



Implementing Stack using Array

?

Implementing Stack using Linked List

?

Why Linked List instead of Array?

?

Infix to Postfix

Infix expression

operator is in-between operands

(a + b)

A + B * C + D

A + B * (C + D)

Postfix expression

operator is in-between operands

(ab+)

ABC*+D+

ABCD+*+

Infix Expression to a Postfix Expression

- Scan input string from left to right character by character.
- If the character is an operand, put it into output stack.
- If the character is an operator and operator's stack is empty, push operator into operators' stack.
- If the operator's stack is not empty, there may be following possibilities.
- If the precedence of scanned operator is greater than the top most operator of operator's stack, push this operator into operator's stack.

Infix Expression to a Postfix Expression

- If the precedence of scanned operator is less than or equal to the top most operator of operator's stack, pop the operators from operator’s stack until we find a low precedence operator than the scanned character. Never pop out ('(') or (')') whatever may be the precedence level of scanned character.
- If the character is opening round bracket ('('), push it into operator's stack.
- If the character is closing round bracket (')'), pop out operators from operator's stack until we find an opening bracket ('(').
- Now pop out all the remaining operators from the operator's stack and push into output stack.

Postfix Evaluation

Infix Expression to a Postfix Expression

- Iterate the expression from left to right
- Keep on storing the operands into a stack.
- Once an operator is received, pop the two topmost elements
- Evaluate them
- Push the result in the stack again.