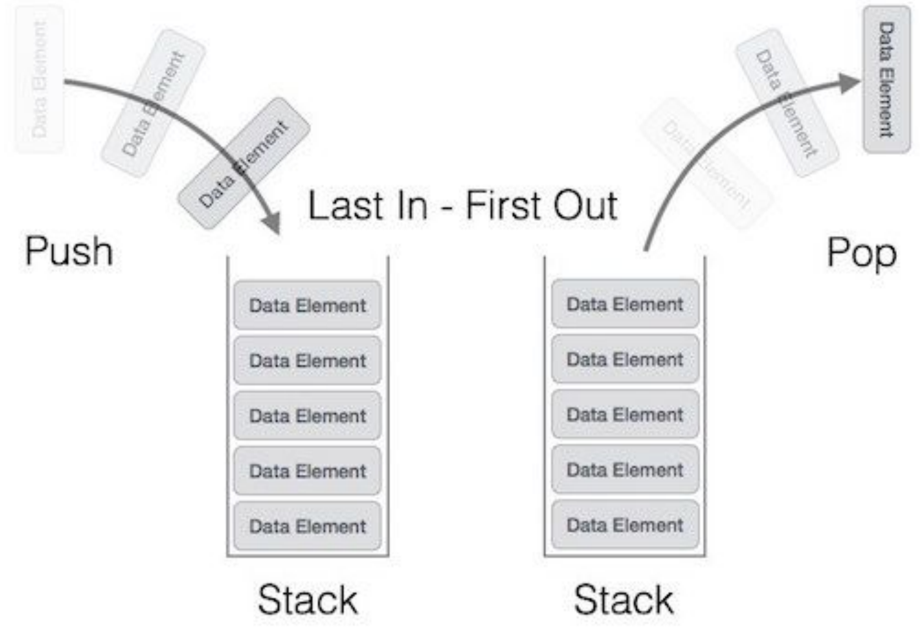
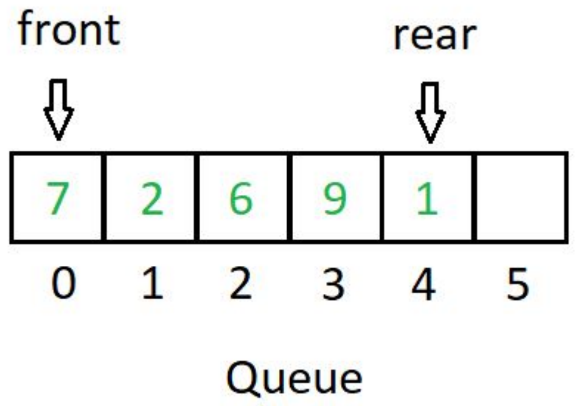
Competitive Programming

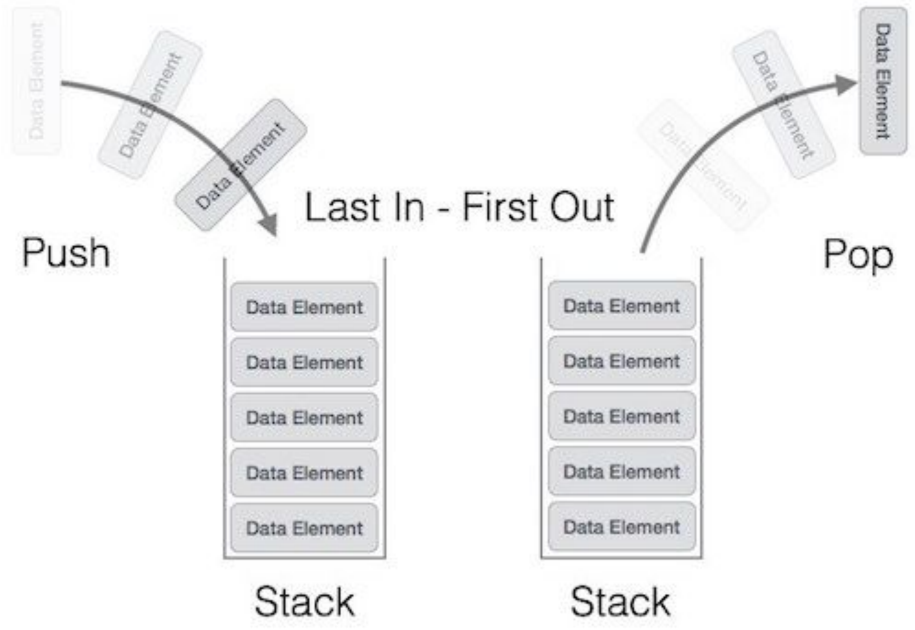
Lec 7

Stack & Queues





Stack



- Which data structure we can use to implement stack?

- Doubly linked list

1. Next pointer
2. Prev pointer
3. size

Stack

Applications:

1. Expression Evaluation/Conversion(syntax tree)

➔ infix

➔ prefix

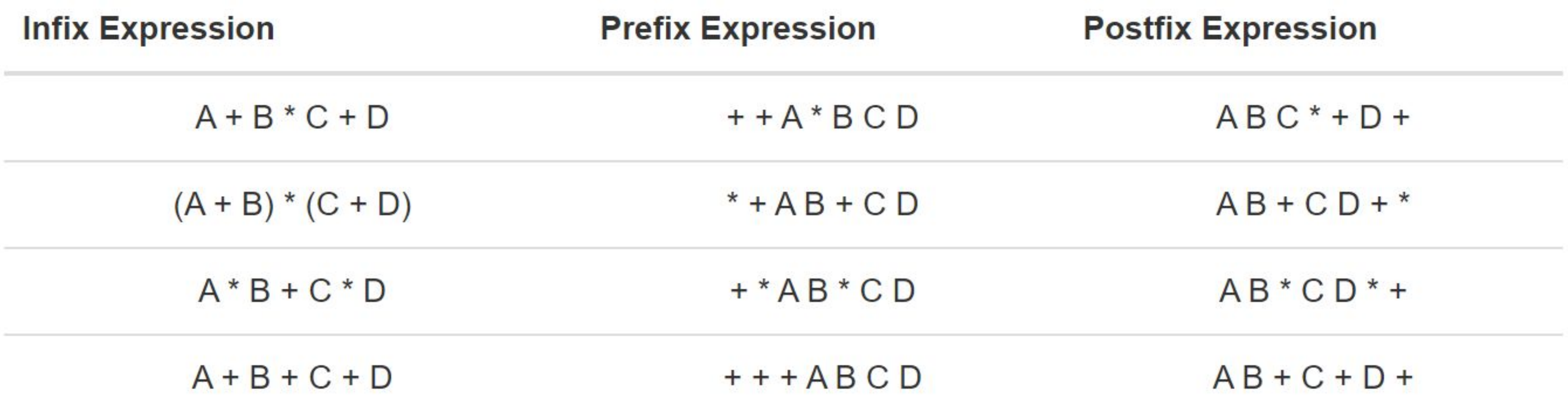
➔ postfix

2. Parenthesis Checking

3. Backtracking/Recursion

4. String Reversal

Stack



Infix Expression	Prefix Expression	Postfix Expression
A + B * C + D	+ + A * B C D	A B C * + D +
(A + B) * (C + D)	* + A B + C D	A B + C D + *
A * B + C * D	+ * A B * C D	A B * C D * +
A + B + C + D	+ + + A B C D	A B + C + D +

Balanced Brackets

A bracket is considered to be any one of the following characters: (,), {, }, [, or].

Two brackets are considered to be a matched pair if the an opening bracket (i.e., (, [, or {) occurs to the left of a closing bracket (i.e.,),], or }) of the exact same type. There are three types of matched pairs of brackets: [], {}, and ().

Sample Input	Sample Output
3	YES
{[()]}	NO
{[(])}	YES
{{[[(())]]}}	

Redundant Braces

Given a string A denoting an expression. It contains the following operators ’+’, ‘-‘, ‘*’, ‘/’.

Check whether A has redundant braces or not.

Return 1 if A has redundant braces, else return 0.

Note: A will be always a valid expression.

Input 1:
    A = "((a + b))"
Output 1:
    1
	Input 2:
	    A = "(a + (a + b))"
	Output 2:
	    0

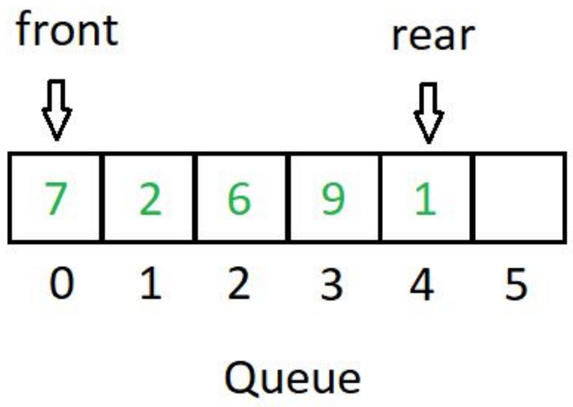
Infix to Postfix

Given an infix expression in the form of a string str. Convert this infix expression to postfix expression.
Precedence order ^,*,/,+,-.

Example:
Input:
```
2
a+b*(c^d-e)^(f+g*h)-i
A*(B+C)/D
```

Output:
```
abcd^e-fgh*+^*+i-
ABC+*D/
```

Queue



- Which data structure we can use to implement stack?
- Doubly linked list

Types of Queue

Queue

- Back for " push "
- Front for " pop "

```
queue < int > q;
```

Doubly ended queue

- Back for both " push " and " pop "
- Front for both " push " and " pop "

```
deque < int > dq;
```

Priority queue

- min/max heap [most important for competitive pro. and interview]

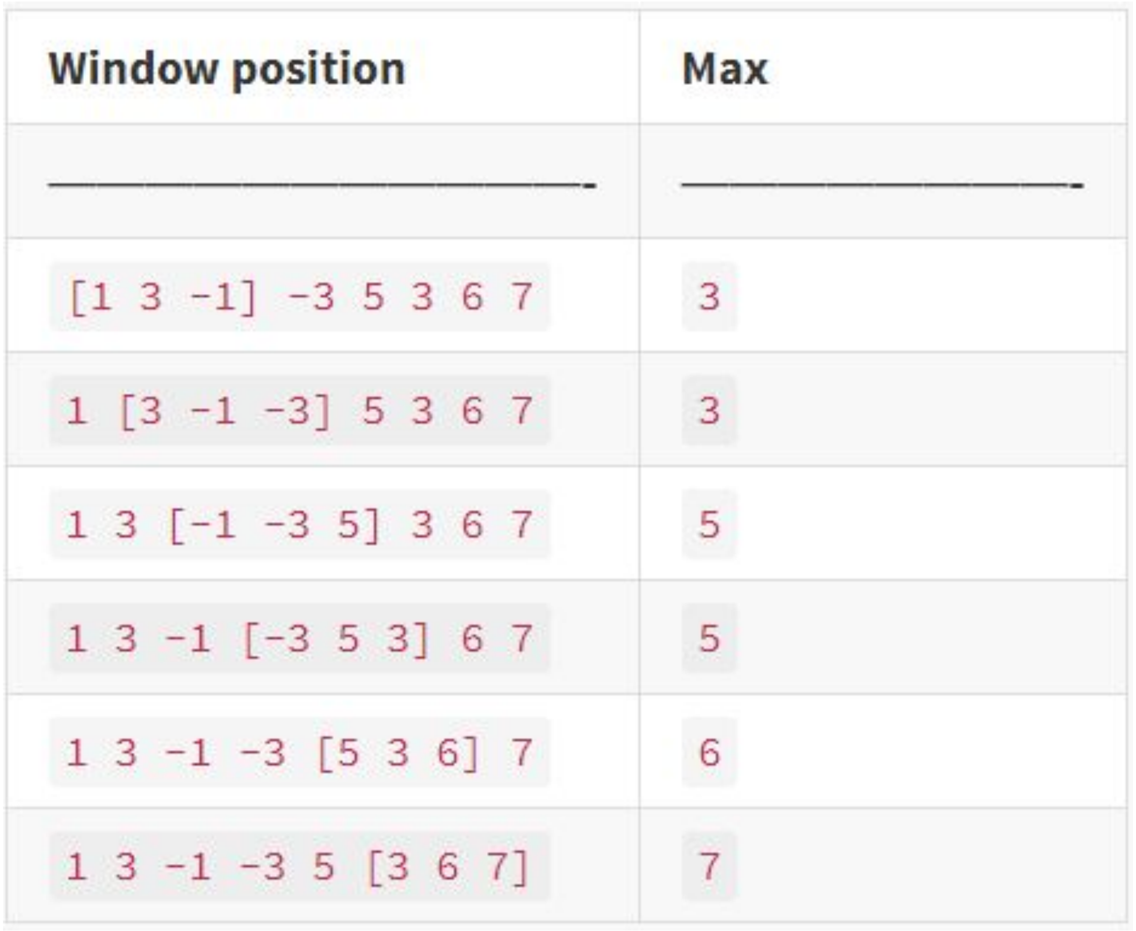
```
priority_queue < int > pq;
```

Sliding Window Maximum

Given an array of integers A. There is a sliding window of size B which
is moving from the very left of the array to the very right.
You can only see the w numbers in the window. Each time the sliding window moves
rightwards by one position. You have to find the maximum for each window.
The following example will give you more clarity.

The array A is [1 3 -1 -3 5 3 6 7], and B is 3.

Sliding Window Maximum



Window position	Max
———————————	———————————
`[1 3 -1] -3 5 3 6 7`	`3`
`1 [3 -1 -3] 5 3 6 7`	`3`
`1 3 [-1 -3 5] 3 6 7`	`5`
`1 3 -1 [-3 5 3] 6 7`	`5`
`1 3 -1 -3 [5 3 6] 7`	`6`
`1 3 -1 -3 5 [3 6 7]`	`7`

Solution Link

Homework

Evaluate Expression	Easy
Simplify Directory Path	Medium
Largest Rectangle in Histogram	Hard

Resources for “ Largest Rectangle in Histogram ”

- https://medium.com/@dimko1/largest-rectangle-in-histogram-bbd7c1e1158
- https://leetcode.com/problems/largest-rectangle-in-histogram/discuss/?currentPage=1&orderBy=hot&query=