**3.1 Stack**

Stack algorithm practice problems:

Easy

1) Parenthesis Checker

2) Reverse a String using Stack

3) Why do we need Prefix and Postfix notations?

4) Reverse an array using Stack

5) How would I modify my code to delete chars from the beginning?

6) Check for balanced parentheses in an expression

7) Reverse a Stack using queue

8) Delete Middle element from stack

9) Reverse individual words

Medium

1. How to implement a queue using two stacks?
2. Implement Stack using Queues
3. Implement a stack using a single queue
4. Evaluate Postfix Expression
5. Next Greater Element
6. Nearest Smaller Element
7. Find next Smaller of next Greater in an array
8. Sort a stack using a temporary stack.
9. Stock Span Problem
10. Reverse a stack using recursion
11. Infix to Postfix Conversion using Stack
12. Implement two stacks in an array
13. Delete consecutive same words in a sequence
14. Lexicographically largest subsequence containing all distinct characters only once
15. Iterative approach to check if a Binary Tree is BST or not
16. Minimize length by removing subsequences forming valid parenthesis from a given string
17. Special Stack
18. Clone a stack without using extra space | Set 2
19. Count of Subarrays whose first element is the minimum
20. Length of the longest valid substring
21. Find index of closing bracket for a given opening bracket in an expression
22. Next Greater Frequency Element
23. Find maximum difference between nearest left and right smaller elements
24. Maximum product of indexes of next greater on left and right

Hard

1. Check if the given permutation is valid stack permutation or not
2. Design a stack that supports getMin() in O(1) time and O(1) extra space
3. Design a data structure that supports insert(), delete(), getRandom() and getMin() in O(1) time complexity
4. Check if two expressions with brackets are same
5. How to efficiently implement k stacks in a single array?
6. Find the largest rectangular area possible in a given histogram
7. Implementation of stack using Doubly Linked List
8. The Celebrity Problem
9. Design custom Browser History based on given operations
10. Maximum size rectangle binary sub-matrix with all 1s | Set 2
11. Sort a stack using another stack
12. Implement a stack that supports findMiddle() and deleteMiddle() operations in O(1) time complexity
13. Maximum people a person can see while standing in a line in both direction
14. Count of distinct differences between two maximum elements of every Subarray
15. Largest Rectangular Area in a Histogram
16. Range Queries for Longest Correct Bracket Subsequence Set
17. Find maximum of minimum for every window size in a given array
18. Print ancestors of a given binary tree node without recursion

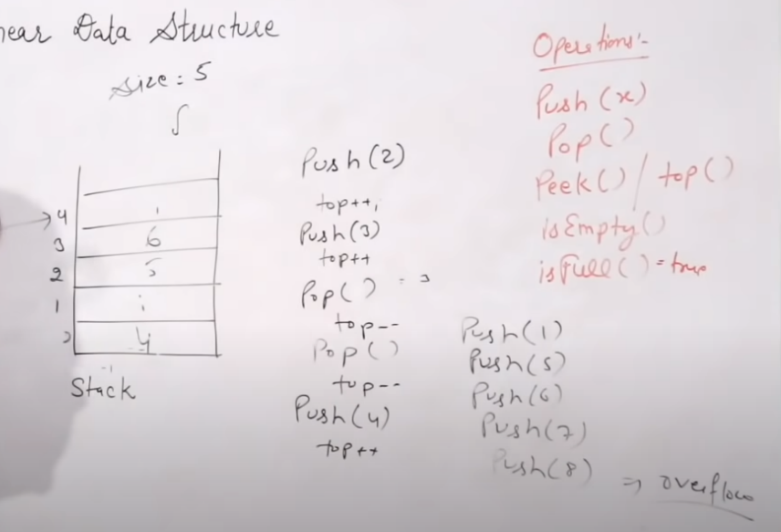
Link: https://www.geeksforgeeks.org/top-50-problems-on-stack-data-structure-asked-in-interviews/

**3.1 Stack in Data Structure | Introduction to Stack**

Stack always follow LIFO (Last In First Out) principle

It is an ordered list or collection list

Here insertion and deletion are performed at one end only and it’s known as top

Fundamental operation in Stack

1. push()
2. pop()
3. peek() \ top()
4. isEmpty()
5. isFull()

Stack implementation.

Static => array

Dynamic => linklist

Stack has two condition

Underflow => when stack is empty

Overflow => when stack is full

Application of Stack

1. Reverse a string.
2. Undo mechanism in text editor (ctr + z)
3. Recursion and function call
4. Check the balance of parenthesis.
5. Infix to postfix or prefix conversion.
6. For the evaluation of post fix expression

3.1 = DONE

**3.2 Implementation of Stack using Array**

1. push()
2. pop()
3. peek() \ top()
4. display()

for this operation the time complexity is O(1).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

3.2 = DONE

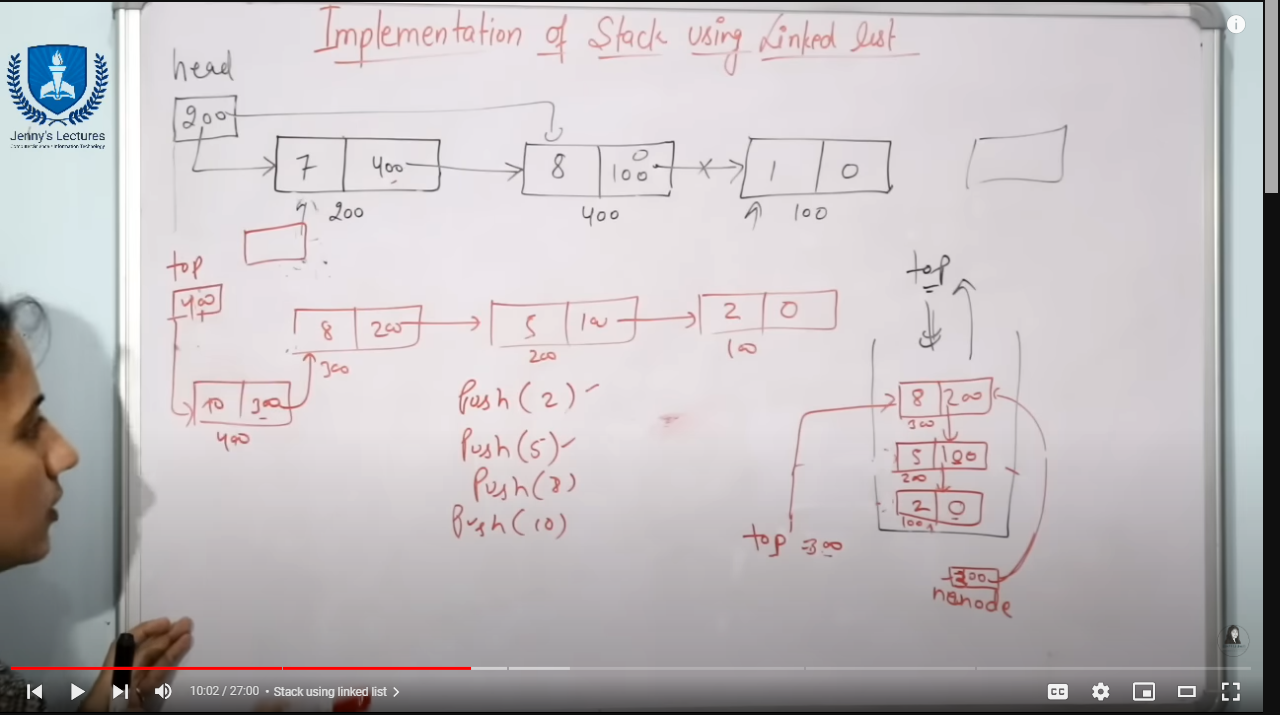
**3.3 Stack implementation using Linked List**

Here insertion and deletion are performed at one end only and it’s known as top

Here in link list the starting can be head or tail (head = top)

Vertically writing the notes as stack

We will always follow the beginning means from head node because of time complexity

 A person writing on a whiteboard

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**Queue**

Easy

1. Reverse First k Elements of Queue
2. Implement a Queue using an Array
3. Print all elements of a queue in a new line
4. Level with maximum number of nodes
5. Breadth First Search or BFS for a Graph
6. Find Minimum Depth of a Binary Tree

Medium

1. Implement a Deque
2. Implement a Circular Queue
3. Check if a queue can be sorted into another queue using a stack
4. Implement Stack using Queues
5. Implement Stack using Two Queues
6. Implement Queue using Two Stacks
7. Design a Queue data structure to get minimum or maximum in O(1) time
8. Check whether a given graph is Bipartite or not
9. Print Right View of a Binary Tree
10. An Interesting Method to Generate Binary Numbers from 1 to n
11. Implement a Queue using a Stack
12. Reverse a queue using recursion
13. Implement Priority Queue using Linked List
14. Implement Queue using Deque
15. Flatten a multilevel linked list
16. Find next right node of a given key
17. Detect cycle in an undirected graph using BFS
18. Minimum steps to reach target by a Knight
19. Islands in a graph using BFS
20. Flood Fill Algorithm
21. Minimum steps to reach target by a Knight
22. First negative integer in every window of size k
23. Level order traversal in spiral form
24. Minimum time required to rot all oranges
25. Queue based approach or first non-repeating character in a stream
26. Shortest distance in a maze
27. Geek in a Maze
28. Find shortest safe route in a path with landmines
29. Find the first circular tour that visits all petrol pumps
30. Connect Nodes at Same Level

Hard

1. Find the first non-repeating character from a stream of characters
2. Maximum of all subarrays of size k using a queue
3. Implement LRU Cache using Queue
4. Design a Queue data structure to get the maximum or minimum of sliding window
5. Find if there is a path between two vertices in a directed graph
6. Design a Data Structure for LRU Cache
7. Trapping Rain Water
8. Maximum cost path from source node to destination
9. Trapping Rain Water
10. Maximum cost path from source node to destination
11. Snake and Ladder Problem
12. Minimum Cost Path in a directed graph via given set of intermediate nodes
13. Turn a Queue into a Priority Queue
14. Interchange elements of Stack and Queue without changing order

Link: https://www.geeksforgeeks.org/top-50-problems-on-queue-data-structure-asked-in-sde-interviews/