

PSG COLLEGE OF TECHNOLOGY
DEPARTMENT OF COMPUTER APPLICATIONS
WORKSHEET-1

1. Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.

Write pseudocode for the following

MinStack() initializes the stack object.

void push(int val) pushes the element val onto the stack.

void pop() removes the element on the top of the stack.

int top() gets the top element of the stack.

int getMin() retrieves the minimum element in the stack.

You must implement a solution with $O(1)$ time complexity for each function.

2. Given two integer arrays pushed and popped each with distinct values, return true if this could have been the result of a sequence of push and pop operations on an initially empty stack, or false otherwise.

Example 1:

Input: pushed = [1,2,3,4,5], popped = [4,5,3,2,1]

Output: true

Explanation: We might do the following sequence:

push(1), push(2), push(3), push(4), pop() -> 4, push(5), pop() -> 5, pop() -> 3, pop() -> 2, pop() -> 1

Example 2:

Input: pushed = [1,2,3,4,5], popped = [4,3,5,1,2]

Output: false

Explanation: 1 cannot be popped before 2.

3. You have a target array of integers and an integer n. There is a stream of integers from 1 to n in ascending order. You want to build a stack that exactly matches the target array in the same order (bottom to top). You start with an empty stack and can perform two operations:

Push: Push the next integer from the stream onto the stack.

Pop: Pop the top integer from the stack.

Follow these rules to build the target stack:

Read integers sequentially from the stream.

For each integer read, you can either:

Push it onto the stack, or

Push it then immediately pop it if it doesn't belong in the target.

Stop once the stack matches the target array from bottom to top.

Return the list of operations ("Push" and "Pop") needed to build the target stack.

4. Implement stack using queues
5. Implement Queue using stacks
6. Implement circular queue
7. Write recursive algorithms for the following:
 - i) Factorial of a number
 - ii) Fibonacci series
 - iii) Binary search
 - iv) Tower of hanoi
8. Write an algorithm to convert an infix expression to postfix expression
9. Write an algorithm rangeCount() that takes a stack of integers and an integer as an argument and returns the number of integers in stack that are atleast a.
10. Formulate an algorithm to evaluate the postfix expression.
11. Write an algorithm to check the well- formedness of parentheses in an expression using stack.
12. Given an array of size N-1 with distinct integers in the range of 1 to N, write an algorithm to find the missing element.
13. Given a queue of integers, how do you reverse the order of the elements of the queue?
14. Write the addressing function of the following types of matrices
 - a) lower triangular matrix
 - b) Toeplitz matrix. Toeplitz matrix is a matrix where the elements on the diagonal are same
 - c) Tridiagonal matrix
15. Write algorithm for the following. Indicate the data structures used in this case.
Check whether the given string is of the syntax wcwr where w is a string with alphabets a and b and wr is the reverse of the string. Ex: w=abb and wr= bba