Competitive Programming and Mathematics Society

Data Structures Workshop

Please ask our helpful programming team if you are confused or stuck! We are happy to help :)

1) **Daily Temperatures**

   Given an array of integers `temperatures` represents the daily temperatures, return an array `answer` such that `answer[i]` is the number of days you have to wait after the `i`th day to get a warmer temperature. If there is no future day for which this is possible, keep `answer[i] = 0` instead.

   Link: https://leetcode.com/problems/daily-temperatures/

2) **Time Needed to Buy Tickets** There are `n` people in line to buy a ticket, with the `i`th person wanting `tickets[i]` tickets. If each person can only buy 1 ticket at a time and then must go back to the end of the line to get another ticket, find the amount of time which the `k`th person must spend waiting before they can get all the tickets they want. Each purchase of a ticket takes 1 second

   Link: https://leetcode.com/problems/time-needed-to-buy-tickets/

3) **Valid Parentheses**

   Given a string s containing just the characters `(`, `)`, `{`, `}`, `[` and `]`, determine if the input string is valid.

   An input string is valid if:

   - Open brackets must be closed by the same type of brackets.
   - Open brackets must be closed in the correct order.
   - Every close bracket has a corresponding open bracket of the same type.

   Link: https://leetcode.com/problems/valid-parentheses/

4) **Evaluate Reverse Polish Notation**

   You are given an array of strings `tokens` that represents an arithmetic expression in Reverse Polish Notation.

   Evaluate the expression. Return an integer that represents the value of the expression.

   - Each token will be an integer or an operator. The valid operators are '+', '-', '*', and '/'.
   - Inputs will be valid and any intermediate results will be storeable in an `int`
   - The input represents a valid arithmetic expression in a reverse polish notation.
   - Truncate division to zero

   Link: https://leetcode.com/problems/evaluate-reverse-polish-notation/

5) **Implement Queue using Stacks**

   Implement a queue using only two stacks. The implemented queue should support all the functions of a normal queue (`push`, `peek`, `pop`, and `empty`).

- You can only use the standard operations of a stack: `push`, `peek`, `pop`, `size` and `empty`

Link: https://leetcode.com/problems/implement-queue-using-stacks/

Extension: How many stacks do you need to implement a double-ended queue (deque)? That is, a queue where you can push or pop from either end of the queue. Can you get an O(1) amortised push and pop from either end?

6) **Shortest Subarray with Sum at least K**

Given an integer array `nums` and an integer `k`, return the length of the shortest non-empty contiguous subarray of `nums` with a sum of at least `k`. If there is no such subarray, return `-1`.

Link: https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/

7) **Sliding Window Maximum**

You are given an array of integers `nums`, there is a sliding window of size `k` which is moving from the very left of the array to the very right. You can only see the `k` numbers in the window. Each time the sliding window moves right by one position.

Return an array which takes each window from left to right and has the corresponding maximum value of that window in the corresponding index in the array. (The question statement on Leetcode is kind of bad, look at the examples).

Link: https://leetcode.com/problems/sliding-window-maximum/

8) **Largest Rectangle in Histogram**

Given an array of integers `heights` representing the histogram's bar height where the width of each bar is 1, return the area of the largest rectangle in the histogram.

Link: https://leetcode.com/problems/largest-rectangle-in-histogram/