Competitive Programming and Mathematics Society

2521 Stacks and Queues

CPMSoc

Welcome

- Mathematics workshops will run every odd-numbered week (3, 5, 7, ...)
- Programming ones will run every even-numbered week (4, 6, 8, ...)
- Slides will be uploaded on our website (unswcpmsoc.com)

Attendance form :D

Workshop Overview

- Stacks
- Queues
- Why use Stacks and Queues?
- Monotonic Stack

Stacks

- What are stacks?

Stacks

- What are stacks?
- (add) Push on the **top**
- (delete) Pop from the **top**
- (read) Peek at the **top** element

Bracket Matching

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.

Queues

- What are queues?

Queues

- What are queues?
- (add) Push to the **back**
- (delete) Pop from the **front**
- (read) Peek at the **front** element

Monotonic Stack

- You may be familiar with the term *"monotonically increasing/decreasing"* from first-year maths.
- If an array of integers is monotonically increasing, then as we go from left to right, the values increase or stay the same.
- A monotonic stack is a mix between a sorted array and a stack! However, it's not the same as a sorted array with LIFO ordering.

Monotonic Stack

We can turn an array into a monotonic stack by removing all the elements that are out of order.
How to create a monotonically increasing stack.

1. Create an empty stack to store the elements.
2. Iterate through the input array from left to right.
3. For each element, do the following:
   a. While the stack is not empty and the current element is greater than the top element of the stack, pop elements from the stack.
   b. Push the current element onto the stack. Continue this process until you have processed all the elements.

Problem - Daily Temperatures

Given an array of integers `temperatures` that represents the daily temperatures, return an array `answer` such that `answer[i]` is the number of days you have to wait after the *ith* day to get a warmer temperature.

If there is no future day for which this is possible, keep `answer[i] == 0` instead.

Solution - Daily Temperatures

- What is an $O(N^2)$ solution to this problem?

Solution - Daily Temperatures

- What is an $O(N^2)$ solution to this problem?
- Can you identify the repeated computations?

Approach

1. Create an empty stack `s` and a `result` array.
2. Iterate through the `temperatures` array. We compare `temperatures[i]` with the current items on the stack `s`.
   a. If the stack has no elements, there is nothing to pop!.
   b. Otherwise ....
      How do we decide which elements to pop, so that the monotonic property is preserved?
   c. How do we calculate `result[i]`?
3. Push `temperatures[i]` onto the monotonic stack!

Attendance form :D

Feedback form :D

Further events

Please join us for:

- Maths workshop next week
- Programming workshop in two weeks