



Kyle Lang <kyle.everett.lang@gmail.com>

Daily Coding Problem: Problem #30 [Medium]

1 message

Daily Coding Problem <founders@dailycodingproblem.com>

Tue, Sep 10, 2019 at 11:37 AM

To: kyle.everett.lang@gmail.com

**Daily Coding Problem**

Good morning! Here's your coding interview problem for today.

This problem was asked by Facebook.

You are given an array of non-negative integers that represents a two-dimensional elevation map where each element is unit-width wall and the integer is the height. Suppose it will rain and all spots between two walls get filled up.

Compute how many units of water remain trapped on the map in $O(N)$ time and $O(1)$ space.

For example, given the input [2, 1, 2], we can hold 1 unit of water in the middle.

Given the input [3, 0, 1, 3, 0, 5], we can hold 3 units in the first index, 2 in the second, and 3 in the fourth index (we cannot hold 5 since it would run off to the left), so we can trap 8 units of water.

[Upgrade to premium](#) and get in-depth solutions to every problem, including this one.

If you liked this problem, feel free to forward it along so they can [subscribe here](#)! As always, shoot us an email if there's anything we can help with!

Ready to interview? [Take Triplebyte's quiz](#) and skip straight to final interviews with top tech companies!

No more? [Unsubscribe](#).

© 2019 Daily Coding Problem. All rights reserved.