NEW YORK INSTITUTE
OF TECHNOLOGY

Shwetha Jayaraj <sjayaraj@nyit.edu>

Daily Coding Problem: Problem #44 [Medium]

1 message

Daily Coding Problem <founders@dailycodingproblem.com>
To: sjayaraj@nyit.edu

Thu, Jul 21, 2022 at 11:37 AM

Daily Coding Problem

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

This problem was asked by Google.

We can determine how "out of order" an array A is by counting the number of inversions it has. Two elements `A[i]` and `A[j]` form an inversion if `A[i] > A[j]` but `i < j`. That is, a smaller element appears after a larger element.

Given an array, count the number of inversions it has. Do this faster than O(N^2) time.

You may assume each element in the array is distinct.

For example, a sorted list has zero inversions. The array [2, 4, 1, 3, 5] has three inversions: (2, 1), (4, 1), and (4, 3). The array [5, 4, 3, 2, 1] has ten inversions: every distinct pair forms an inversion.

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!

Master algorithms together on Binary Search. Create a room, invite your friends, and race to finish the problem!

No more? Snooze or unsubscribe.