Problem 1826: Faulty Sensor

Problem Information

Difficulty: Easy
Acceptance Rate: 50.29%
Paid Only: Yes
Tags: Array, Two Pointers

Problem Description

An experiment is being conducted in a lab. To ensure accuracy, there are**two** sensors collecting data simultaneously. You are given two arrays `sensor1` and `sensor2`, where `sensor1[i]` and `sensor2[i]` are the `ith` data points collected by the two sensors.

However, this type of sensor has a chance of being defective, which causes **exactly one** data point to be dropped. After the data is dropped, all the data points to the **right** of the dropped data are **shifted** one place to the left, and the last data point is replaced with some **random value**. It is guaranteed that this random value will **not** be equal to the dropped value.

* For example, if the correct data is `[1,2,_**3**_ ,4,5]` and `3` is dropped, the sensor could return `[1,2,4,5,_**7**_]` (the last position can be **any** value, not just `7`).

We know that there is a defect in **at most one** of the sensors. Return _the sensor number (_`1` _or_`2` _) with the defect. If there is**no defect** in either sensor or if it is**impossible** to determine the defective sensor, return _`-1` _._

Example 1:

Input: sensor1 = [2,3,4,5], sensor2 = [2,1,3,4] **Output:** 1 **Explanation:** Sensor 2 has the correct values. The second data point from sensor 2 is dropped, and the last value of sensor 1 is replaced by a 5.

Example 2:

Input: sensor1 = [2,2,2,2,2], sensor2 = [2,2,2,2,5] **Output:** -1 **Explanation:** It is impossible to determine which sensor has a defect. Dropping the last value for either sensor could produce the output for the other sensor.

Example 3:

Input: sensor1 = [2,3,2,2,3,2], sensor2 = [2,3,2,3,2,7] **Output:** 2 **Explanation:** Sensor 1 has the correct values. The fourth data point from sensor 1 is dropped, and the last value of sensor 1 is replaced by a 7.

Constraints:

* `sensor1.length == sensor2.length` * `1 <= sensor1.length <= 100` * `1 <= sensor1[i], sensor2[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
int badSensor(vector<int>& sensor1, vector<int>& sensor2) {

}
};
```

Java:

```
class Solution {
public int badSensor(int[] sensor1, int[] sensor2) {

}
}
```

Python3:

```
class Solution:
def badSensor(self, sensor1: List[int], sensor2: List[int]) -> int:
```