Problem 3468: Find the Number of Copy Arrays

Problem Information

Difficulty: Medium
Acceptance Rate: 46.51%
Paid Only: No
Tags: Array, Math

Problem Description

You are given an array `original` of length `n` and a 2D array `bounds` of length `n x 2`, where `bounds[i] = [ui, vi]`.

You need to find the number of **possible** arrays `copy` of length `n` such that:

1. `(copy[i] - copy[i - 1]) == (original[i] - original[i - 1])` for `1 <= i <= n - 1`. 2. `ui <= copy[i] <= vi` for `0 <= i <= n - 1`.

Return the number of such arrays.

Example 1:

Input: original = [1,2,3,4], bounds = [[1,2],[2,3],[3,4],[4,5]]

Output: 2

Explanation:

The possible arrays are:

* `[1, 2, 3, 4]` * `[2, 3, 4, 5]`

Example 2:

Input: original = [1,2,3,4], bounds = [[1,10],[2,9],[3,8],[4,7]]

Output: 4

Explanation:

The possible arrays are:

* `[1, 2, 3, 4]` * `[2, 3, 4, 5]` * `[3, 4, 5, 6]` * `[4, 5, 6, 7]`

Example 3:

Input: original = [1,2,1,2], bounds = [[1,1],[2,3],[3,3],[2,3]]

Output: 0

Explanation:

No array is possible.

Constraints:

* `2 <= n == original.length <= 105` * `1 <= original[i] <= 109` * `bounds.length == n` * `bounds[i].length == 2` * `1 <= bounds[i][0] <= bounds[i][1] <= 109`

Code Snippets

C++:

```
class Solution {
public:
int countArrays(vector<int>& original, vector<vector<int>>& bounds) {

}
};
```

Java:

```
class Solution {
public int countArrays(int[] original, int[][] bounds) {
```

```
}
}
```

Python3:

```
class Solution:
def countArrays(self, original: List[int], bounds: List[List[int]]) -> int:
```