Problem 3656: Determine if a Simple Graph Exists

Problem Information

Difficulty: Medium
Acceptance Rate: 47.52%
Paid Only: Yes
Tags: Array, Binary Search, Graph, Sorting, Prefix Sum

Problem Description

You are given an integer array `degrees`, where `degrees[i]` represents the desired degree of the `ith` vertex.

Your task is to determine if there exists an **undirected simple** graph with **exactly** these vertex degrees.

A **simple** graph has no self-loops or parallel edges between the same pair of vertices.

Return `true` if such a graph exists, otherwise return `false`.

Example 1:

Input: degrees = [3,1,2,2]

Output: true

Explanation:

![](https://assets.leetcode.com/uploads/2025/08/12/screenshot-2025-08-13-at-24347-am.png)

One possible undirected simple graph is:

* Edges: `(0, 1), (0, 2), (0, 3), (2, 3)` * Degrees: `deg(0) = 3`, `deg(1) = 1`, `deg(2) = 2`, `deg(3) = 2`.

Example 2:

Input: degrees = [1,3,3,1]

Output: false

Explanation:

* `degrees[1] = 3` and `degrees[2] = 3` means they must be connected to all other vertices. * This requires `degrees[0]` and `degrees[3]` to be at least 2, but both are equal to 1, which contradicts the requirement. * Thus, the answer is `false`.

Constraints:

* `1 <= n == degrees.length <= 10■■■■■■■5` * `0 <= degrees[i] <= n - 1`

Code Snippets

C++:

```
class Solution {
public:
bool simpleGraphExists(vector<int>& degrees) {

}
};
```

Java:

```
class Solution {
public boolean simpleGraphExists(int[] degrees) {

}
}
```

Python3:

```
class Solution:
def simpleGraphExists(self, degrees: List[int]) -> bool:
```