Problem 2508: Add Edges to Make Degrees of All Nodes Even

Problem Information

Difficulty: Hard
Acceptance Rate: 34.31%
Paid Only: No
Tags: Hash Table, Graph

Problem Description

There is an **undirected** graph consisting of `n` nodes numbered from `1` to `n`. You are given the integer `n` and a **2D** array `edges` where `edges[i] = [ai, bi]` indicates that there is an edge between nodes `ai` and `bi`. The graph can be disconnected.

You can add **at most** two additional edges (possibly none) to this graph so that there are no repeated edges and no self-loops.

Return `true` _if it is possible to make the degree of each node in the graph even, otherwise return_`false` _._

The degree of a node is the number of edges connected to it.

Example 1:

![](https://assets.leetcode.com/uploads/2022/10/26/agraphdrawio.png)

Input: n = 5, edges = [[1,2],[2,3],[3,4],[4,2],[1,4],[2,5]] **Output:** true **Explanation:** The above diagram shows a valid way of adding an edge. Every node in the resulting graph is connected to an even number of edges.

Example 2:

![](https://assets.leetcode.com/uploads/2022/10/26/aagraphdrawio.png)

Input: n = 4, edges = [[1,2],[3,4]] **Output:** true **Explanation:** The above diagram shows a valid way of adding two edges.

Example 3:

![](https://assets.leetcode.com/uploads/2022/10/26/aaagraphdrawio.png)

Input: n = 4, edges = [[1,2],[1,3],[1,4]] **Output:** false **Explanation:** It is not possible to obtain a valid graph with adding at most 2 edges.

Constraints:

* `3 <= n <= 105` * `2 <= edges.length <= 105` * `edges[i].length == 2` * `1 <= ai, bi <= n` * `ai != bi` * There are no repeated edges.

Code Snippets

C++:

```
class Solution {
public:
bool isPossible(int n, vector<vector<int>>& edges) {

}
};
```

Java:

```
class Solution {
public boolean isPossible(int n, List<List<Integer>> edges) {

}
}
```

Python3:

```
class Solution:
def isPossible(self, n: int, edges: List[List[int]]) -> bool:
```