There are N rooms and you start in room 0.  Each room has a distinct number in 0, 1, 2, ..., N-1, and each room may have some keys to access the next room.

Formally, each room i has a list of keys rooms[i], and each key rooms[i][j] is an integer in [0, 1, ..., N-1] where N = rooms.length.  A key rooms[i][j] = v opens the room with number v.

Initially, all the rooms start locked (except for room 0).

You can walk back and forth between rooms freely.

Return true if and only if you can enter every room.

**Example 1:**

**Input:** [[1],[2],[3],[]]

**Output:** true

**Explanation:**

We start in room 0, and pick up key 1.

We then go to room 1, and pick up key 2.

We then go to room 2, and pick up key 3.

We then go to room 3. Since we were able to go to every room, we return true.

**Example 2:**

**Input:** [[1,3],[3,0,1],[2],[0]]

**Output:** false

**Explanation:** We can't enter the room with number 2.

**Note:**

1. 1 <= rooms.length <= 1000
2. 0 <= rooms[i].length <= 1000
3. The number of keys in all rooms combined is at most 3000.

有 N 个房间，开始时你位于 0 号房间。每个房间有不同的号码：0，1，2，...，N-1，并且房间里可能有一些钥匙能使你进入下一个房间。

在形式上，对于每个房间 i 都有一个钥匙列表 rooms[i]，每个钥匙 rooms[i][j] 由 [0,1，...，N-1] 中的一个整数表示，其中 N = rooms.length。 钥匙 rooms[i][j] = v 可以打开编号为 v 的房间。

最初，除 0 号房间外的其余所有房间都被锁住。

你可以自由地在房间之间来回走动。

如果能进入每个房间返回 true，否则返回 false。

**示例 1:**

**输入:** [[1],[2],[3],[]]

**输出:** true

**解释:**

我们从 0 号房间开始，拿到钥匙 1。

之后我们去 1 号房间，拿到钥匙 2。

然后我们去 2 号房间，拿到钥匙 3。

最后我们去了 3 号房间。

由于我们能够进入每个房间，我们返回 true。

**示例 2:**

**Input:** [[1,3],[3,0,1],[2],[0]]

**Output:** false

**Explanation:** 我们不能进入 2 号房间。

**Note:**

1. 1 <= rooms.length <= 1000
2. 0 <= rooms[i].length <= 1000
3. 所有房间中的钥匙数量总计不超过 3000。