Problem 841: Keys and Rooms

Problem Information

Difficulty: Medium
Acceptance Rate: 75.21%
Paid Only: No
Tags: Depth-First Search, Breadth-First Search, Graph

Problem Description

There are `n` rooms labeled from `0` to `n - 1` and all the rooms are locked except for room `0`. Your goal is to visit all the rooms. However, you cannot enter a locked room without having its key.

When you visit a room, you may find a set of **distinct keys** in it. Each key has a number on it, denoting which room it unlocks, and you can take all of them with you to unlock the other rooms.

Given an array `rooms` where `rooms[i]` is the set of keys that you can obtain if you visited room `i`, return `true` _if you can visit**all** the rooms, or_ `false` _otherwise_.

Example 1:

Input: rooms = [[1],[2],[3],[]] **Output:** true **Explanation:** We visit room 0 and pick up key 1. We then visit room 1 and pick up key 2. We then visit room 2 and pick up key 3. We then visit room 3. Since we were able to visit every room, we return true.

Example 2:

Input: rooms = [[1,3],[3,0,1],[2],[0]] **Output:** false **Explanation:** We can not enter room number 2 since the only key that unlocks it is in that room.

Constraints:

* `n == rooms.length` * `2 <= n <= 1000` * `0 <= rooms[i].length <= 1000` * `1 <= sum(rooms[i].length) <= 3000` * `0 <= rooms[i][j] < n` * All the values of `rooms[i]` are

unique.

Code Snippets

C++:

```
class Solution {
public:
bool canVisitAllRooms(vector<vector<int>>& rooms) {

}
};
```

Java:

```
class Solution {
public boolean canVisitAllRooms(List<List<Integer>> rooms) {

}
}
```

Python3:

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