Problem Title: Count Friend Groups (Asked by Twitter)

Friend Groups in a Classroom

This problem was asked by **Twitter**.

Scenario & Description:

Imagine a classroom of `N` students. Students can be friends with one another, and this friendship relationship is mutual (i.e., if A is a friend of B, B is also a friend of A).

We're given these relationships as an **adjacency list**, where each key is a student and the values are the list of students they're directly friends with.

A **friend group** is a set of students where every student is **connected (directly or indirectly)** to every other student in that set. In other words, we want to find the **connected components** in an undirected graph where students are nodes and friendships are edges.

Your task is to **count the total number of friend groups**.

Input Format:

- An integer `N`, the total number of students (0 to N-1)
- A dictionary `friendship` where:
  - Key = student ID
  - Value = list of friend student IDs

Output Format:

- An integer: total number of friend groups in the class

Examples:

Example 1:

```
Input:
N = 7
friendship = {
  0: [1, 2],
  1: [0, 5],
  2: [0],
  3: [6],
  4: [],
  5: [1],
  6: [3]
}

Output:
3

Explanation:
Group 1: {0, 1, 2, 5}
Group 2: {3, 6}
Group 3: {4}
```

Example 2:

```
Input:
N = 5
friendship = {
  0: [1],
  1: [0, 2],
  2: [1],
  3: [4],
  4: [3]
}

Output:
2

Explanation:
Group 1: {0, 1, 2}
Group 2: {3, 4}
```

Approach:

We treat the problem as a **graph traversal** problem to count **connected components**.

1. Use **DFS** or **BFS** to explore each component.
2. Keep a `visited` set to track already processed students.
3. For every unvisited student, start a new DFS/BFS and increment the group count.

✅ Sample Python Code:

```python
def count_friend_groups(N, friendship):
    visited = set()
    groups = 0

    def dfs(student):
        for friend in friendship.get(student, []):
            if friend not in visited:
                visited.add(friend)
                dfs(friend)

    for student in range(N):
        if student not in visited:
            visited.add(student)
            dfs(student)
            groups += 1

    return groups

# Example
N = 7
friendship = {
  0: [1, 2],
  1: [0, 5],
  2: [0],
  3: [6],
  4: [],
  5: [1],
  6: [3]
}

print(count_friend_groups(N, friendship))  # Output: 3
```

Practice Links:

- LeetCode: Number of Provinces *(very similar)*
- GFG: Find the number of islands *(graph traversal concept)*

Video Explanation (Recommended):

- Graph Connected Components - DFS (YouTube)
- Leetcode 547: Number of Provinces