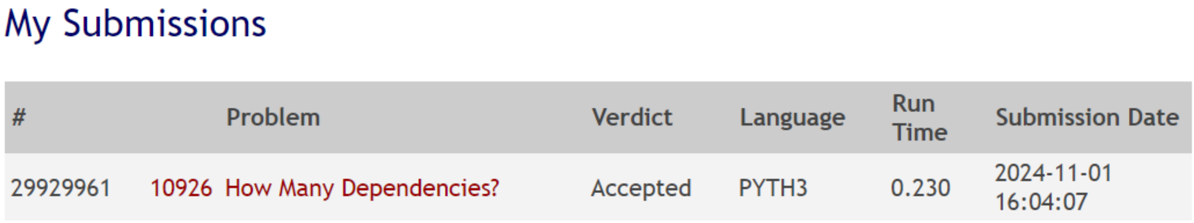
Recorrido: BFS

```
from collections import deque, defaultdict


def bfs(task, graph):
    visited = set()
    queue = deque([task])
    count = 0

    while queue:
        current = queue.popleft()
        for neighbor in graph[current]:
            if neighbor not in visited:
                visited.add(neighbor)
                queue.append(neighbor)
                count += 1

    return count

def find_task_with_most_dependencies(scenario):
    n = scenario[0]

    graph = defaultdict(list)
    for i in range(n):
        task_info = scenario[i + 1]
        dependencies = task_info[1:]
        task_id = i + 1
        for dep in dependencies:
            graph[task_id].append(dep)


    max_dependencies = -1
    best_task = -1
    for task in range(1, n + 1):
        dependencies_count = bfs(task, graph)
        if dependencies_count > max_dependencies or (dependencies_count ==
max_dependencies and task < best_task):
            max_dependencies = dependencies_count
            best_task = task

    return best_task

results = []
while True:
    n = int(input().strip())
    if n == 0:
        break
    scenario = [n]
    for _ in range(n):
        line = list(map(int, input().strip().split()))
        scenario.append(line)
    results.append(find_task_with_most_dependencies(scenario))

for result in results:
    print(result)
```



My Submissions

#	Problem	Verdict	Language	Run Time	Submission Date
29929961	10926 How Many Dependencies?	Accepted	PYTH3	0.230	2024-11-01 16:04:07