

E Noon's Architecture Audit

TIME LIMIT: 1.0s
MEMORY LIMIT: 256MB



The devops team at Noon is auditing their massive backend codebase to improve system stability. The system is composed of N different "Services" (Because hisham hates the term microservices), labeled from 1 to N .

The architecture defines dependencies where one service inherits functionality from another. These relationships can be modeled as a Directed Acyclic Graph (DAG). An arrow from Service X to Service Y means X inherits from Y .

The team is worried about the "Diamond Dependency" problem. This occurs if there is a redundancy in the data flow. Specifically, a Diamond Dependency exists if there are two services, X and Y , such that there are at least two different inheritance paths from X to Y .

A path is defined as a sequence of services C_1, C_2, \dots, C_k where C_i inherits from C_{i+1} . Your task is to help the Noon engineering team by writing a program that analyzes the dependency graph and determines if a Diamond Dependency exists.

INPUT

The first line contains an integer T ($1 \leq T \leq 10$) — the number of test cases. For each test case:

The first line contains an integer N ($1 \leq N \leq 1000$) — the number of services.

The following N lines describe the inheritance graph.

The i -th line starts with an integer M_i ($0 \leq M_i < N$), representing the number of services that Service i inherits from.

This is followed by M_i distinct integers, representing the parent services of Service i .

OUTPUT

For each test case, print "Yes" (without quotes) if the architecture contains a Diamond Dependency, and "No" otherwise.

SAMPLES

Sample input 1	Sample output 1
2 4 2 2 3 1 4 1 4 0 3 1 2 1 3 0	Yes No