

Homework 6, CSE 232

Due October 8th Note: On most of the problem sets through the semester, I'll put a horizontal line with "Optional" under it. Any problems below this section are encouraged - I think they're interesting and will help you learn the subject - but not necessary to complete in order to get credit for the homework.

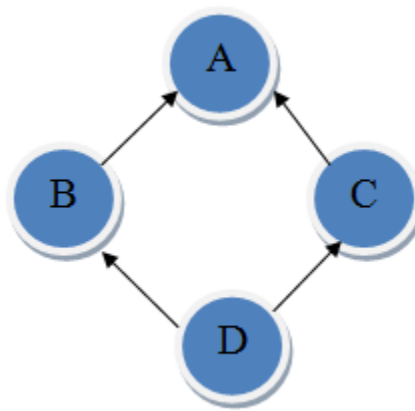
Problem 1

This week's problem is [GCJ Round 1C 2012 - Diamond Inheritance](#).

Here's the problem statement:

You are asked to help diagnose class diagrams to identify instances of diamond inheritance. The following example class diagram illustrates the property of diamond inheritance. There are four classes: A, B, C and D. An arrow pointing from X to Y indicates that class X inherits from class Y.

```
In [1]: from IPython.display import Image, display
display(Image(filename='./week6_hwing.png'))
```



In this class diagram, D inherits from both B and C, B inherits from A, and C also inherits from A. An inheritance path from X to Y is defined as a sequence of classes $X, C_1, C_2, C_3, \dots, C_n, Y$ where X inherits from C_1 , C_i inherits from C_{i+1} for $1 \leq i \leq n-1$, and C_n inherits from Y. There are two inheritance paths from D to A in the example above. The first path is D, B, A and the second path is D, C, A.

A class diagram is said to contain a diamond inheritance if there exists a pair of classes X and Y such that there are at least two different inheritance paths from X to Y. The above class diagram is a classic example of diamond inheritance. Your task is to determine whether or not a given class diagram contains a diamond inheritance.

Input

The first line of the input gives the number of test cases, T. T test cases follow, each specifies a class diagram. The first line of each test case gives the number of classes in this diagram, N. The classes are numbered from 1 to N. N lines follow. The i th line starts with a non-negative integer M_i indicating the number of classes that class i inherits from. This is followed by M_i distinct positive integers each from 1 to N representing those classes. You may assume that:

If there is an inheritance path from X to Y then there is no inheritance path from Y to X. A class will never inherit from itself.

Output

For each diagram, output one line containing “Case #x: y”, where x is the case number (starting from 1) and y is “Yes” if the class diagram contains a diamond inheritance, “No” otherwise.

Limits

$$1 \leq T \leq 50.$$

$$0 \leq M_i \leq 10.$$

Small dataset

$$1 \leq N \leq 50.$$

Large dataset

$$1 \leq N \leq 1,000.$$

a) *How can we represent the class inheritance as a graph? What are the nodes of the graph? What are the edges of the graph? We talked about several different ways to classify graphs. Classify the graph of class inheritance according to these distinctions. Is this graph directed or undirected, weighted or unweighted, simple or non-simple, and cyclic or acyclic?*

b) *Compare the maximum number of vertices and edges in the graph under the Small and Large input limits. How much memory would be required to represent this graph when stored as either an adjacency matrix or an adjacency list in the Large input limits?*

c) *Suppose we pick a starting vertex and try to traverse the graph from that point. How will the vertices we reach be related to the starting vertex? Will this depend on whether we use a breadth-first or a depth-first search?*

d) *How many times will any end vertex Y show up in your search from a particular start vertex X? How does this depend on the inheritance paths between X and Y? Write a solution to the problem using a modified depth-first or breadth-first search and attach it to your homework. Don't forget to run your code on the Small and Large inputs to make sure it works.*

Optional No optional work this week.