Bonus: Isomorphism (30%)

|  |
| --- |
| Program: iso.(py|cpp|java)  Input: iso.in  Output: iso.out |

**Description**

Given two graphs and . Use a backtracking approach to check whether the two graphs are isomorphic or not. Recall that two graphs are isomorphic iff there is a one-to-one function such that nodes are adjacent in iff are adjacent in .

Input

The first line on the input file contains an integer denoting the number of test cases. In each test case, you are given two adjacency matrices representing graphs .   
The test case starts with four positive integers on a separate line followed by adjacency matrix representing graph and followed by adjacency matrix representing graph .

Output

For each test case, print on a separate line “Yes” if the two graphs are isomorphic and “No” otherwise.

Additional deliverables

1. Time and space analysis for the algorithm part only.
2. Is there any kind of input that we can easily conclude that the two graphs are **NOT isomorphic?**
3. Is there any kind of input that we can easily conclude that the two graphs are **isomorphic?**
4. What is the maximum depth of the backtracking tree for any given input?

**Sample Input /output**

iso.out

3

3 3 3 3

0 1 1

1 0 0

1 0 0

0 0 1

0 0 1

1 1 0

4 4 4 4

0 1 1 1

1 0 1 1

1 1 0 1

1 1 1 0

0 0 1 1

0 0 1 1

1 1 0 1

1 1 1 0

2 2 3 3

1 0

0 1

0 1 1

1 0 1

1 1 0

Yes

No

No

iso.in