Problem B
László Babai

Time limit: 1 second
Memory limit: 256 megabytes

Problem Description

> László Babai is a Hungarian computer scientist and mathematician. He is a Gödel prize winner and an outstanding researcher in the fields of the theory of computation, algorithms, combinatorics, and group theory. Last year, he proposed a subexponential-time algorithm solving Graph Isomorphism in $\exp((\log n)^{O(1)})$-time, and the best previous result is an $\exp(O(\sqrt{n \log n}))$-time algorithm.

Graph Isomorphism is a famous NP problem in theoretical computer science, however, you may wonder what it is. Let us explain for a bit. Given two undirected graphs $A = (V_A, E_A)$ and $B = (V_B, E_B)$, where A's vertex set is $V_A = \{a_1, a_2, a_3, \ldots, a_{n_A}\}$, and B's vertex set is $V_B = \{b_1, b_2, b_3, \ldots, b_{n_B}\}$. Graph A and B are isomorphic if and only if

1. A and B have the same amount of vertices and edges,
2. There exists a bijective (one-to-one and onto) function $f : V_A \to V_B$ such that $\{u, v\} \in E_A$ if and only if $\{f(u), f(v)\} \in E_B$.

In other words, we can relabel the vertex set of graph A to obtain graph B.

Graph Isomorphism is still neither known to be in P nor NP-complete. As up and coming computer scientists, we must be ambitious and never be afraid to dream big! Therefore, let us take on the challenge of testing if two 3-vertex undirected simple graphs G_1 and G_2 are isomorphic and show the world that we too can accomplish something.

Input Format

The first line of the input will be a single integer T ($T \leq 100$) representing the number of test cases that will follow.

Every test case then starts with the number of edges m ($0 \leq m \leq 3$) in the first undirected simple graph of 3 vertices (numbered from 1 to 3), followed by m lines each containing two distinct integers u, v ($u \neq v$, $u, v \in \{1, 2, 3\}$) indicating that there exists an edge between vertex u and v. You may assume that there is at most one edge between any pair of vertices. After that the description of the second graph follows in the same format.

Output Format

If the two graphs are isomorphic than output "`yes`" on one line. If not, output "`no`" instead.

Sample Input

```
3
3
1 2
2 3
3 1
3
1 3
2 1
3 2
2
1 2
1 3
0
1
2 3
1
1 2
```

Sample Output

```
yes
no
yes
```