

beecrowd | 1610

Dudu Faz Serviço

Por Rafael Regis, Universidade de São Paulo - São Carlos  Brazil**Timelimit: 1**

Dudu precisa de um documento para finalizar uma tarefa em seu trabalho. Após pesquisar um pouco, ele descobre que este documento depende de outros documentos que, por sua vez, necessitam de outros documentos e assim por diante.

Dudu chegou a uma lista final com todos os documentos que deverá precisar. Com essa lista em mãos, ele suspeita que a mesma possui loops. Por exemplo, se um documento A depende do documento B que por sua vez depende do documento A, tornaria a tarefa interminável. Veja que neste caso o loop tem apenas dois documentos, pode haver loops com três ou mais!

Dada a lista das dependências entre os documentos, ajude Dudu a saber se um dia conseguirá todos os documentos, ou seja, se não existe um loop na lista.

Entrada

Na primeira linha você terá um inteiro **T** ($T = 100$) indicando o número de casos de teste.

Na primeira linha de cada caso teremos os números inteiros **N** ($2 \leq N \leq 100^*$ ou $2 \leq N \leq 104^{**}$) e **M** ($1 \leq M \leq 300^*$ ou $1 \leq M \leq 3 \cdot 104^{**}$), indicando o número de documentos e as dependências existentes. Em cada uma das **M** linhas seguintes, terão dois inteiros **A** ($1 \leq A$) e **B** ($B \leq N$, com $A \neq B$), indicando que o documento **A** depende do documento **B**. Pode haver dependências repetidas!

*Ocorre em aproximadamente 90% dos casos de teste;

**Ocorre nos demais casos de teste.

Saída

Para cada caso, imprima SIM caso exista pelo menos um *loop* e NAO caso contrário.

Exemplo de Entrada	Exemplo de Saída
3	NAO
2 1	SIM
1 2	SIM
2 2	
1 2	
2 1	
4 4	
2 3	
3 4	
4 2	
1 3	