# Graph Generator

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

DreamGrid has made a graph generator. The generator will take in a single integer $n$ and generate an undirected graph with $n$ vertices.

More specifically, the generator will generate an empty graph $G$ and a permutation $p$ of $\{1, 2, \ldots, n\}$ . After that, the generator will do the following operations $n$ times and during the $q$-th operation:

1. find the all connected components $C_1, C_2, \ldots, C_m$ in $G$.
2. choose a subset $S_q$ of $\bigcup_{i=1}^{m} C_i$ such that no two vertices in $S_q$ belong to the same connected component.
3. create a new vertex with index $p_q$ in $G$ and add an edge between $p_q$ and every vertex $v$ in $\bigcup_{i \in S_q} C_{bel_i}$, where $bel_i$ is the index of connected component which $i$ belongs to.

Given the final generated graph, DreamGrid would like to know all the $p_q$ and $S_q$.

## Input

There are multiple test cases. The first line of input contains an integer $T$, indicating the number of test cases. For each test case:

The first line contains two integers $n$ and $m$ ($1 \le n \le 10^5, 0 \le m \le \min(10^5, \frac{n(n-1)}{2})$) – the number of vertices and the number of edges.

Each of the next $m$ lines contains two integers $u_i$ and $v_i$ ($1 \le u_i, v_i \le n$). There will be no self loops or multiple edges.

It is guaranteed that neither the sum of all $n$ nor the sum of all $m$ exceeds $2 \times 10^6$.

## Output

For each test case, if the the graph cannot be generated by the generator, output "No" in the first line. Otherwise, output "Yes" in the first line. Then in the $i$-th of the following line, output two integers $q_i$ and $s_i$ ($1 \le q_i \le n$) followed with $s_i$ integers: $a_1, a_2, \ldots, a_{s_i}$ ($1 \le a_j \le n$) – the index of the newly created vertex and the subset during the $i$-th operation. If there are multiple solutions, print any of them.

## Example

| standard input | standard output |
|---|---|
| 3<br>3 0<br>4 4<br>1 2<br>2 3<br>3 4<br>2 4<br>5 5<br>1 2<br>2 3<br>3 4<br>4 5<br>2 4 | Yes<br>1 0<br>2 0<br>3 0<br>Yes<br>1 0<br>4 0<br>3 1 4<br>2 2 1 3<br>No |