Challenge NPC II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Tajian is solving the following problem:

Given a graph with n points, the initial graph has no edges. Tajian has some edges to add to the graph, and he has divided these edges into k groups, where the i-th group contains c_i edges. For each $x \in [1, n]$, Tajian wants to know the minimum number of groups of edges he needs to select from these k groups to add to the graph in order to ensure that the number of connected components in the graph does not exceed x.

Although it is not immediately obvious, there are papers that indicate that the minimum coloring number of a spanning tree is an NPC problem. However, Tajian does not know this, so he wants you to help him solve this problem.

Input

A single test case contains multiple data sets.

The first line of input is the number of data sets T ($1 \leq T \leq 10^3$), representing the number of data sets in this test case.

For each data set, the first line contains two integers n, k ($2 \leq n \leq 16$, $1 \leq k \leq 200$), representing the number of points and the number of groups of edges.

The next section describes the information for the k groups of edges. For the i-th group of edges, the first line contains an integer c_i ($1 \leq c_i \leq 200$), representing the number of edges in this group. The following c_i lines each contain two integers u, v ($1 \leq u, v \leq n, u \neq v$), representing an edge in this group.

For a single data set, it is guaranteed that $\sum c_i \leq 200$. For all data, it is guaranteed that $\sum 2^n \leq 2^{16}$.

Output

For each data set, output a line of n integers, where the i-th integer represents the answer when $x = i$. If it is impossible to ensure that the number of connected components does not exceed i under any circumstances, output "-1".

Example

standard input	standard output
4 3 3 1 1 2 1 2 3 1 1 3 4 3 1 2 3 2 1 2 1 3 1 3 4 6 2 3 1 2 3 4 5 6 2 2 3 4 5 4 2 1 2 3 2 1 3 1 2	2 1 0 2 1 1 0 2 2 1 1 1 0 -1 1 1 0