

11929 Ambiguous Forests

The latest assignment in your graph theory course asks you to find the largest collection of edges in a graph which does not contain a cycle. However, you didn't attend the class when the assignment was given so you've asked to see the notes of two of your friends. Unfortunately, one of your friends must have written down the wrong graph because their two graphs are different. In fact, the only thing the two graphs seem to have in common is that they both include the same number of edges m and the edges are numbered from 0 to $m - 1$.



Since you don't know which graph is correct, you have two options. On one hand, you could flip a coin to guess which graph is correct. On the other hand, you could find the largest subset T of the integers from 0 to $m - 1$ so that in each of the two graphs, the edges which are numbered with an integer in T are acyclic. Hoping to guarantee some partial marks, you choose the second option.

Input

The first integer k denotes the number of test cases. Each test case begins with three integers n_1, n_2, m where n_1 denotes the number of nodes in the first graph, n_2 denotes the number of nodes in the second graph, and m denotes the number of edges in both graphs.

Then m lines follow where the i -th such line consists of four numbers u_1, v_1, u_2, v_2 where $0 \leq u_j < v_j < n_j$ for each $j = 1, 2$. This means (u_1, v_1) is an edge in the first graph and (u_2, v_2) is an edge in the second graph.

No graph will have more than 50 nodes and the number of edges is at most 200. Furthermore, no graph will have two of the same edge.

Output

The output for each test case is a single line beginning with an integer t denoting the size of the largest subset T of integers 0 to $m - 1$ such that neither graph contains a cycle where all edges are numbered with an integer in T . Following this, you should output the numbers of the edges of such a subset in increasing order. Say these integers are $e_1 < e_2 < \dots < e_t$. If there are multiple subsets of integers of size t that contain no cycle in either graph, then output the one that minimizes e_t . If there are still multiple such subsets with the same minimum value for e_t , output the one among these that minimizes e_{t-1} , and so on.

Sample Input

```
2
3 4 3
0 1 0 1
1 2 1 2
0 2 2 3
```

```
5 5 5
0 1 0 1
1 2 0 2
1 3 2 4
2 3 3 4
2 4 1 2
```

Sample Output

```
2 0 1
4 0 2 3 4
```