

DAGCH: Graph Challenge

题目描述

厨师 Hawlader 和厨师 Heickal 是好朋友。除了烹饪以外，他们还喜爱算法。Hawlader 喜欢图论而 Heickal 喜欢数论。

又一次，厨师 Hawlader 给厨师 Heickal 讲授图论。“嘿，Heickal，你应该更专注于对图论的学习。数论中还剩下什么呢？图论是这个世界的全部，你甚至可以将数论问题划归到图论来解决，你懂吗？”Hawlader 说。而对于 Heickal 来说，他并不厌恶图论，根据她的理论，生活应该是数论，图论，动态规划，数据结构，ad hoc 等等等等的混合。要成为一名优秀的厨师，你必须知道所有这些东西而不仅仅只有图论。

- “哦，你是图论的大师？那你知道 DFS？”Heickal 问道。
- “当然。DFS，深度优先搜索，这是一个基本图论算法。”Hawlader 略显愠色的回答道。
- “因此，对于任意一幅图，你可以按照 DFS 进入的时间给它们编号？”
- “太简单了。用下面的伪代码就行了”

```
int C = 1;
void DFS(int u)
{
    new_number[u] = C;
    C++;

    // initially all value of visited array is set to false
    visited[u] = true;

    // here v can be chosen in an arbitrary order
    for each v such that there is a edge from u to v
    if(visited[v] == false)
        DFS(v);
}
```

- “好的，我可以给你一个难一点的问题了。”Heickal 带着坏笑的说道。
- “什么？放马过来吧！”Hawlader 笑嘻嘻的回答道。
- “我将会给你一个有 N 个节点， M 条边的有向图，每个节点的编号是它们 DFS 时进入时刻的编号。并且，从 1 号结点出发，所有的结点都可达。”
- “好的，那问题呢？”

- “等等，我给你一些定义，一个结点 x 被称为是另一个结点 y 的 **supreme vertex**，如果存在一条有向路 $x = v_0, v_1, \dots, v_k = y$ ，满足 $x \leq y \leq v_i$ ，对于所有 $0 \leq i \leq k$ 。也就是说，一条从 x 出发到 y 的有 0 个或者更多个中间结点的有向路，满足所有中间结点的编号都大于 x 和 y 的编号，并且 x 的编号小于 y 。一个结点 v 被称为是另一个结点 w 的 **superior vertex**，如果 v 是 w 所有的 **supreme vertex** 中，编号最小的。你将得到 Q 个询问。每个询问，将会给定一个结点 v ，你需要找出有多少结点，将 v 视为其 **superior vertex**。”
- “噢！呃...”

厨师 Hawlader 无法解决这个问题，他正在向你求助。

输入格式

输入数据的第一行有一个整数 T ，表示数据的组数。每组测试数据的第一行，有三个整数 N, M, Q 。接下来的 M 行，每行有一对整数 U_i 和 V_i 表示一条从 U_i 出发，指向 V_i 的有向边。接下来的一行包含 Q 个用空格隔开的整数， P_1, P_2, \dots, P_Q 。每个整数表示一组询问。

输出格式

对于每组询问，输出一行表示对应的结果。

数据限制

- $1 \leq T \leq 10$
- $2 \leq N \leq 10^5$
- $N - 1 \leq M \leq 2 \times 10^5$
- $1 \leq Q \leq 10^5$
- $1 \leq U_i, V_i \leq N, U_i \neq V_i$
- 没有重边，所有结点接从 1 号结点可达。

样例数据

输入

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

```
1 2 3
8 9 8
1 2
1 7
2 3
2 5
3 4
5 6
7 8
6 4
8 4
1 2 3 4 5 6 7 8
```

输出

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

样例解释

样例一中，结点 3 有且仅有结点 1 作为其 **supreme vertex** 和 **superior vertex**。因为有一条 1 到 3 的有向边。结点 2 有一个 **supreme vertex** 1 (1 \rightarrow 3 \rightarrow 2)。因为结点 2 仅有这一个 **supreme vertex**，所以这个结点也是其 **superior vertex**。

样例二中，结点 4 有两个 **supreme vertex** 1(1 \rightarrow 7 \rightarrow 8 \rightarrow 4) 和 2(2 \rightarrow 5 \rightarrow 6 \rightarrow 4)，因此结点 1 是结点 4 的 **superior vertex**。

Problem Setter: Shiplu Hawlader

Problem Tester: Hiroto, Gerald Agapov

Translated by : xiaodao