There are N cities and M roads in *Dragons Country*. Cities are numbered 1 through N, and each road connects exactly two cities. There are totally K dragons D_1, D_2, ..., D_K in these N cities. Dragon D_i lives in city C_i and has initially S_i heads. It grows N_i new heads every minute while *alive*! A dragon is alive if it has positive number of heads.

We want to hire a number of warriors to kill all dragons. We specify the initial city of each warrior. Then on each minute, each warrior can either go through a road from his current city to an adjacent city, or select an alive dragon in his current city and cut off one of its heads. We can specify the strategy of each warrior on each minute, and after they are done, any alive dragon D_i will grow its N_i new heads.

We want to find the minimum number of warriors with which all dragons can be killed in a finite amount of time.

Input

There are multiple test cases in the input. For each test case, the first line contains three space-separated integers N, M and K ($1 \leq N \leq 300$, $0 \leq M \leq N(N-1)/2$, and $1 \leq K \leq 1000$). Each of the next M lines contains two integers a and b ($1 \leq a \neq b \leq N$) indicating a road between cities a and b. The i-th line of the next K lines describes dragon D_i with three space-separated integers C_i, S_i and N_i ($1 \leq C_i \leq N$, $1 \leq S_i \leq 10^5$, $0 \leq N_i \leq 10^5$). The input terminates with a line containing '`0 0 0`' which should not be processed.

Output

For each test case, output a line containing the minimum number of warriors who can eventually kill all dragons.

Sample Input

```
2 1 1
1 2
1 7 4
4 4 2
1 2
2 4
4 1
1 3
1 2 3
2 3 1
0 0 0
```

Sample Output

```
5
2
```