Problem E. Teyvat

Time limit: 8 seconds
Memory limit: 256 Megabytes

Yoshinow2001 has not logged in the GenshinImpact for a long time, and there are many more Dendroculus on the map of the Teyvat continent.

The map of the GenshinImpact can be seen as an **undirected connected graph** $G = (V, E)$ of n points, m edges, where each vertex $v \in V$ represents a location on the map, and each edge $e = (v_s, v_t)$ represents a road from the point v_s to v_t.

Since Yoshinow2001 has a lot of Dendroculus did not take, so he will give you a total of Q queries, each query indicates that there are k vertexs $\{a_1, a_2, \ldots, a_k\}$ on the graph G, these vertexs need to take Dendroculus. He wants to know how many vertex pairs (S, T) satisfy $1 \leq S \leq T \leq n$ so that any simple path from S to T passes through all vertexs in the set $\{a_1, \ldots, a_k\}$ exactly once.

Input

Each test contains multiple test cases.The first line of input contains a single integer $T(1 \leq T \leq 1 \times 10^4)$—-the number of test cases.The description of test cases follows.

The first line contains three integers $n, m, Q(1 \leq n \leq 5 \times 10^5, 1 \leq m, Q \leq 1 \times 10^6)$ correspondingly represent the number of vertexs, the number of edges, the number of queries.

The following m lines contains two integers $u_i, v_i(u_i \neq v_i)$ indicating an undirected edge between u_i and v_i. It is guaranteed that $\forall i, j : 1 \leq i < j \leq m$ satisfy $(u_i, v_i) \neq (u_j, v_j)$.

The following Q lines. Each line begins with an integer k, representing the number of vertexs queried. Next k integers $a_1, \ldots, a_k$, representing the vertexs.

It is guaranteed that the sum of n for each test cases does not exceed 1.5×10^6, the sum of $m, Q, \sum_{i=1}^{Q} k_i$ for each test cases does not exceed 3×10^6.

Output

For each test case, output Q lines, where the i-th line contains an integer, representing the answer to the i-th query.

Notes: In this problem, you may need more stack space to pass this problem. We suggest you to add the following code into your main function if you use C++.

```cpp
int main() {
    int size(512<<20); // 512M
    __asm__ ( "movq %0, %%rsp\n"::"r"((char*)malloc(size)+size));
    // YOUR CODE
    ...
    exit(0);
}
```

And if you use the code above please **DON'T forget to add exit(0)**; in the end of your main function, otherwise your code may get RE.

Example

standard input	standard output
1	3
3 3 3	1
1 3	0
3 2	
1 2	
1 1	
2 2 3	
3 1 2 3	