NOI '23 P3 - Depth First Search

Time limit: 6.0s **Memory limit:** 512M

Depth-first search is a common search algorithm. Using this algorithm, we can obtain a tree T from an undirected connected graph $G = (V, E)$ with no self-loops nor parallel edges, and a certain starting point s.

The algorithm can be described as follows:

- Set the stack S to be empty, and let $T = (V, \emptyset)$, which means that the edge set of T is initially empty.
- First, push the starting point s into S.
- Visit the top vertex u of the stack and mark u as "visited".
- If there is a vertex v adjacent to u and not yet visited, arbitrarily select one from these vertices and let (u, v) be added to the edge set of T. Then, push v into the stack S, and go back to step 3. If there is no such vertex, pop u out of the stack.

It can be proved that when G is a connected graph, the algorithm will obtain a certain spanning tree T of G. However, the tree T obtained by the algorithm may not be unique, depending on the search order, i.e., the vertex selected in step 4. If a specific search order can be chosen so that the tree obtained by the algorithm is exactly T, then we call T an s-dfs tree of G with respect to the starting point s.

Now, given a tree T with n vertices labeled from 1 to n, and an additional m edges, we guarantee that these m edges are distinct and connect different vertices, and are different from the $n - 1$ tree edges in T. We call these additional m edges non-tree edges. Among these n vertices, we specify exactly k vertices as special vertices.

Now, you want to know how many ways there are to select a subset of these m non-tree edges (you can possibly select none) such that: after the tree edges of T and the selected non-tree edges are combined to form a graph G, there exists a special vertex s such that T is an s-dfs tree of G.

Since the answer may be very large, you only need to output the number of solutions modulo $(10^9 + 7)$.

Input Specification

The first line of input contains an integer c, which represents the test case number. $c = 0$ represents that this test case is a sample test.

The second line of input contains three positive integers n, m, k, which represent the number of vertices, the number of non-tree edges, and the number of critical points, respectively.

Then $n - 1$ lines follow, each containing two positive integers u, v, representing a tree edge of T. It is guaranteed that these $n - 1$ edges form a tree.

Then m lines follow, each containing two positive integers a, b, representing a non-tree edge. It is guaranteed that (a, b) does not coincide with an edge on the tree and there are no duplicate edges.

The last line of input contains k positive integers $s_1, s_2, \ldots, s_k$, representing the labels of the k special vertices. It is guaranteed that $s_1, s_2, \ldots, s_k$ are distinct from each other.

Output Specification

Output a line containing an integer, representing the number of solutions, taken modulo $(10^9 + 7)$.

Sample Input 1

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

Sample Output 1

```
3
```

Explanation for Sample Output 1

In this sample, there are three ways to select the non-tree edges: selecting only the edge $(1, 3)$, selecting only the edge $(2, 4)$, or not selecting any non-tree edges. If we select only the edge $(1, 3)$, or do not select any non-tree edges, we can show that T is a 3-dfs tree of G. The specified search order is as follows:

- Put 3 into the stack S. At this time, $S = [3]$.
- Mark 3 as "visited".
- Since 3 is adjacent to 2 and 2 is "unvisited", put 2 into the stack S and add $(3, 2)$ to tree T. At this time, $S = [3, 2]$.
- Mark 2 as "visited".
- Since 2 is adjacent to 1 and 1 is "unvisited", put 1 into stack S and add $(2, 1)$ to tree T. At this time, $S = [3, 2, 1]$.
- Since all the vertices adjacent to 1 are "visited", pop 1 off the stack. At this time, $S = [3, 2]$.
- Since all the vertices adjacent to 2 are "visited", pop 2 off the stack. At this time, $S = [3]$.
- Since 3 is adjacent to 4 and 4 is "unvisited", put 4 into stack S and add $(3, 4)$ to tree T. At this time, $S = [3, 4]$.
- Since all the vertices adjacent to 4 are "visited", pop 4 off the stack. At this time, $S = [3]$.
- Since all the vertices adjacent to 3 are "visited", pop 3 off the stack and S becomes empty again.

If we select only the edge $(2, 4)$, we can show that T is a 2-dfs tree of G. The specified search order is as follows:

- Put 2 into stack S. At this time, $S = [2]$.
- Mark 2 as "visited".
- Since 2 is adjacent to 3 and 3 is "unvisited", put 3 into the stack S, and add $(2, 3)$ to tree T. At this time, $S = [2, 3]$.
- Mark 3 as "visited".

- Since 3 is adjacent to 4 and 4 is "unvisited", put 4 into the stack S, and add $(3, 4)$ to tree T. At this time, $S = [2, 3, 4]$.
- Since all the neighboring vertices of 4 are "visited", pop 4 out of the stack. At this time, $S = [2, 3]$.
- Since all the neighboring vertices of 3 are "visited", pop 3 out of the stack. At this time, $S = [2]$.
- Since 2 is adjacent to 1 and 1 is "unvisited", put 1 into the stack S, and add $(2, 1)$ to tree T. At this time, $S = [2, 1]$.
- Since all the neighboring vertices of 1 are "visited", pop 1 out of the stack. At this time, $S = [2]$.
- Since all the neighboring vertices of 2 are "visited", pop 2 out of the stack and S becomes empty again.

Additional Samples

Sample inputs and outputs can be found here.

- Sample 2 (`ex_dfs2.in` and `ex_dfs2.ans`) corresponds to test cases 4-6.
- Sample 3 (`ex_dfs3.in` and `ex_dfs3.ans`) corresponds to test cases 10-11.
- Sample 4 (`ex_dfs4.in` and `ex_dfs4.ans`) corresponds to test cases 12-13.
- Sample 5 (`ex_dfs5.in` and `ex_dfs5.ans`) corresponds to test cases 14-16.
- Sample 6 (`ex_dfs6.in` and `ex_dfs6.ans`) corresponds to test cases 23-25.

Problem Constraints

For all test data, it is guaranteed that: $1 \le k \le n \le 5 \cdot 10^5, 1 \le m \le 5 \cdot 10^5$.

<table>
<tr><th>Test ID</th><th>$n \le$</th><th>$m \le$</th><th>$k \le$</th><th>Additional Constraints</th></tr>
<tr><td>$1 \sim 3$</td><td>15</td><td>15</td><td>n</td><td rowspan="3">None</td></tr>
<tr><td>$4 \sim 6$</td><td>6</td><td>6</td><td rowspan="2">6</td></tr>
<tr><td>$7 \sim 9$</td><td rowspan="4">300</td><td rowspan="4">300</td></tr>
<tr><td>$10 \sim 11$</td><td rowspan="7">n</td><td>A</td></tr>
<tr><td>$12 \sim 13$</td><td>B</td></tr>
<tr><td>$14 \sim 16$</td><td>None</td></tr>
<tr><td>$17 \sim 18$</td><td rowspan="3">$2 \cdot 10^5$</td><td rowspan="3">$2 \cdot 10^5$</td><td>A</td></tr>
<tr><td>$19 \sim 21$</td><td>B</td></tr>
<tr><td>22</td><td rowspan="2">None</td></tr>
<tr><td>$23 \sim 25$</td><td>$5 \cdot 10^5$</td><td>$5 \cdot 10^5$</td></tr>
</table>

Additional Constraint A: It is guaranteed that in T, vertex i is connected to vertex $i + 1$ $(1 \le i < n)$.

Additional Constraint B: It is guaranteed that if the edges of T are combined with all m non-tree edges to form a graph G, then T is an 1-dfs tree of G.