Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Exercise – *Shopping Trip*

Rick needs to do the shopping for the whole week this afternoon. He wants to visit a number of stores, but he wants to go home after visiting each one to drop off what he bought. Aditionally, he does not want to use the same street twice while visiting the different stores, lest it will be too boring. He is now looking at the map and wondering if it is possible at all. He clearly needs your help.

Problem You are given a map of the city specified as a graph with street intersections as vertices and streets as edges. Aditionally you know the locations of the stores Rick wants to visit. Stores are always located on street intersections. Find out if it is possible for Rick to visit all stores and return home between each successive visits. He can use the same street while going to one store and returning from the same store, but not while on the way to or from different stores. Through intersections he can go as many times as he wants. Also note that two streets can cross without intersecting as there are over- and under-passes.

Input The first line of the input contains T, the number of the test cases. Each of T subsequent lines will contain one testcase. A testcase starts with n, m and s, $(1 \leqslant n, 0 \leqslant m, 1 \leqslant s < n)$ the numbers of vertices (intersections), edges (streets) and stores that need to be visited respectively. Following this come s numbers specifying the store locations given by intersection numbers. Rick's house is always at intersection 0 (the first intersection) and no store is at this intersection. Note that there might be more than one store at an intersection. The following $2m$ numbers specify which intersections are connected by streets, $2i$-th and $2i + 1$-st numbers specifing end points of i-th street. Each street connects two different intersections and no pair of intersections is connected by more than one street.

Output The output should contain a single line for every test case. The line should contain single word `yes` or `no`, depending on whether it is, or is not, possible to solve the problem for that test case.

Sample Input
```
2
6 7 3 3 4 5 0 1 0 2 0 3 1 3 2 3 3 4 3 5
4 4 3 1 2 3 0 1 0 2 2 3 1 3
```

Sample Output
```
yes
no
```