beCP
2022
Task 2.3: Souvenirs (souvenirs)

Authors: Damien Galant, Pierre Vandenhove
Time limit: 1 s   Memory limit: 512 MB

You are a tour guide in beCP-town. Today, you are guiding a tourist who is excited to explore the town. The town is made of n intersections, numbered from 0 to $n-1$, connected by m one-way streets. Your tour starts at intersection s and you want to lead the tourist to intersection t where your friend's souvenir shop is located. You know that k intersections are particularly boring. For these intersections, the tourist will simply follow you to the next intersection you choose, following a one-way street. However, the other intersections are so magnificent that the tourist will not pay attention to your instructions and may go anywhere, following a one-way street. *Can you guarantee that the tourist will reach the souvenir shop?*

Remark: the tourist has a lot of energy and will only agree to stop his tour at the souvenir shop. It is guaranteed that there is at least one street leaving every intersection. The tour continues as long as the souvenir shop is not reached.

Input

The first line of the input consists of three integers n, m, k denoting respectively the number of intersections, one-way streets, and boring intersections.

The next line of the input consists of two integers s, t denoting the indices of the starting intersection and of the intersection of the souvenir shop.

The following m lines contain two integers u_i and v_i, meaning that there is a street from intersection u_i to intersection v_i.

The last line of the input contains k integers b_j denoting the indices of the boring intersections.

Output

Print "YES" (without the quotes) if you can guarantee that the tourist will visit the souvenir shop, "NO" otherwise.

General limits

- $1 \leq n \leq 10^5$;

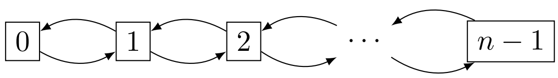
- $1 \leq m \leq 5 \cdot 10^5$;
- $0 \leq k \leq n$;
- $0 \leq s, t < n$;
- $0 \leq u_i, v_i, b_j < n$.

Note: there may be several streets connecting the same pair of intersections, and there may be streets from an intersection to itself.

Additional constraints

Subtask	Points	Constraints
A	10	“line town” (see explanation below), $k = n$
B	15	“line town”
C	15	$k = n$
D	15	$k = 0$
E	20	$n \leq 150$, $m \leq 300$
F	25	No additional constraint

Subtasks A and B only contain *“line town”s*. A “line town” with n intersections has $m = 2 \cdot (n-1)$ streets connecting intersections as follows: for every i such that $0 \leq i < n-1$, there is a street from intersection i to intersection $i+1$ and a street from intersection $i+1$ to intersection i. We depict a “line town” with n intersections in the following drawing.



Example 1

sample1.in
```
3 4 2
0 2
0 1
1 0
1 2
2 1
0 2
```

sample1.out
```
NO
```

There is only one outgoing street from the starting intersection 0, to intersection 1. Intersection 1 is magnificent, and the tourist may choose to always go back to intersection 0. Hence, you cannot make sure that the souvenir shop at intersection 2 is reached. This sample is a "line town", which could be asked in subtasks B, E, and F.

Example 2

sample2.in

```
4 6 1
0 3
0 1
0 2
1 3
2 0
2 3
3 3
0
```

sample2.out

```
YES
```

You start at intersection 0 and may go to 1 or 2. If you go to 2, the tourist may always go back to 0, so this will not guarantee that the souvenir shop at 3 is reached. However, if you go to 1, the tourist has only one possibility in 1, which is to go to 3. Hence, the souvenir shop is reached. This input does not correspond to a "line town".