NEWBEGINNING

After finally managing to choose a single pair of cofounders from among all the acceptable pairs, and graduating (the easier of the two achievements), you're ready to begin another episode of your life.

Your next task is to find an office to work out of. To ensure a short commute, you also decide that you're willing to move and live close to the office.

The city can be represented as a **connected undirected** graph with N vertices labelled $1, \cdots, N$ and M edges of **unit weight**. Exactly K of the vertices are vacant properties, and any one of them would serve as a fine location for either the office or your new home, but not both.

Thus, your task is to find two vertices, A and B, such that the **shortest distance** between A and B is **minimized**.

To ensure you do not have 4999900137 choices like you did when choosing cofounders (you counted!), you decide on a tie-breaking mechanism. Among all possible pairs with shortest distance, you wish to **minimize the index of the office, and among those, minimize the index of your new home**.

Input

The first line contains three space-separated integers, N, M, and K.

The next M lines contain the edges. The $(i+1)$-th ($1 \leq i \leq M$) line contains two space-separated integers a_i and b_i, which are the endpoints of the i-th edge. It is guaranteed that $1 \leq a_i, b_i \leq N$ and $a_i \neq b_i$.

The final line contains K space-separated integers $s_1, \cdots, s_K$, which are the labels of the vacant properties. It is guaranteed that these K integers will be distinct, and $1 \leq s_j \leq N$ for all $1 \leq j \leq N$.

Output

Two space-separated integers - the location of the office and of your new home - which are the lexicographically smallest pair of vacant properties with minimum distance between them.

Constraints

In all test cases, $1 \leq N, M \leq 3 \cdot 10^5, 2 \leq K \leq 3 \cdot 10^5$. Beyond the sample input, the tests are divided into batches with additional constraints. Time limits below are for C/C++; Ocaml gets 2x, Java 3x, and Python 10x.

- 9 points worth satisfy $K = 2, M \leq 10^5$. TL: 200ms.
- 30 points worth satisfy $M \cdot K \leq 3 \cdot 10^5$. TL: 200ms.
- 16 points worth satisfy $M \leq 10^5$. TL: 300ms.
- 35 points worth satisfy no further constraints. TL: 1000ms.
- 10 points worth are given for particularly tricky cases. These all satisfy $N, M, K \leq 10$. TL: 100ms.

Sample explanation

Both pairs $(1, 2)$ and $(1, 3)$ have the minimal distance of 1 between them, so the tie-breaking procedure dictates that the answer is $(1, 2)$.

View submissions (https://cs124.seas.harvard.edu/problem/NEWBEGINNING/code-submission)

Test cases

Input	Output	Points	Timeout

`2 4`
`4 3`
`3 1`
`1 2 3`
	`1 2`	0	100 ms

Hidden	*Hidden*	3	200 ms
Hidden	*Hidden*	3	200 ms
Hidden	*Hidden*	3	200 ms
Hidden	*Hidden*	10	200 ms
Hidden	*Hidden*	10	200 ms
Hidden	*Hidden*	10	200 ms
Hidden	*Hidden*	4	300 ms
Hidden	*Hidden*	4	300 ms
Hidden	*Hidden*	4	300 ms
Hidden	*Hidden*	4	300 ms
Hidden	*Hidden*	7	800 ms
Hidden	*Hidden*	7	800 ms
Hidden	*Hidden*	7	800 ms
Hidden	*Hidden*	7	800 ms
Hidden	*Hidden*	7	800 ms
Hidden	*Hidden*	2	100 ms
Hidden	*Hidden*	2	100 ms
Hidden	*Hidden*	2	100 ms
Hidden	*Hidden*	2	100 ms
Hidden	*Hidden*	2	100 ms

Download (https://cs124.seas.harvard.edu/problem/NEWBEGINNING/test-cases)