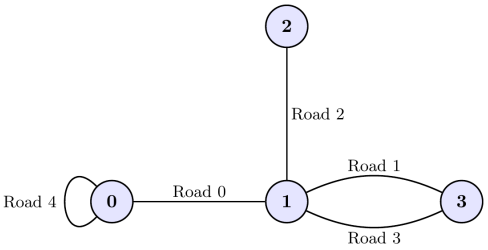
Walks

There are N cities, numbered from 0 to $N-1$. There are also M roads, numbered from 0 to $M-1$. Each road is bidirectional and connects two cities (not necessarily distinct). For each road, we also know its length, which is always equal to 1. It is possible to reach any city from any other city by travelling along a sequence of roads.

A walk from city 0 to city $N-1$ is a sequence of road indices that can be traversed in that order, starting at city 0 and ending at city $N-1$. A road connecting two distinct cities may be traversed in either direction. A road connecting a city to itself may be traversed from that city back to the same city. Walks are considered different if their road-index sequences differ in at least one position; in particular, if multiple roads connect the same pair of cities, they are considered distinct.



Let P be the infinite sequence of all walks from city 0 to city $N-1$, sorted by increasing length. Walks with the same length are sorted lexicographically by their road-index sequences. A walk is lexicographically smaller than another if, at the first position where their road-index sequences differ, it uses the smaller road index. Let $L(P[i])$ denote the length of walk $P[i]$, where P is indexed from 0.

Given K, write a program that computes the following value:

$$\sum_{i=0}^{K-1} L(P[i]) \pmod{10^9+7}$$

Implementation Details

You should implement the following procedure:

```
int path_sum(int N, int M, int K, int[] U, int[] V)
```

- N: the number of cities.
- M: the number of roads.
- K: the number of first walks whose lengths must be summed.
- U, V: arrays of length M describing the roads. For each i such that $0 \leq i < M$, road i connects cities $U[i]$ and $V[i]$.
- This procedure should return the required sum modulo $10^9 + 7$.
- This procedure is called exactly once for each testcase.

Constraints

- $3 \leq N \leq 50\,000$
- $N - 1 \leq M \leq 50\,000$
- $1 \leq K \leq 10^9$
- $0 \leq U[i], V[i] < N$ for each i such that $0 \leq i < M$.
- It is possible to reach any city from any other city by travelling along a sequence of roads.
- Multiple roads may connect the same pair of cities, and a road may connect a city to itself.

Subtasks

Subtask	Score	Additional Constraints
1	6	$K = 1$
2	5	$N \leq 5$, $M \leq 5$, $K \leq 5$
3	10	$K \leq 2$
4	11	$K \leq 5$
5	21	$N \leq 1000$, $M \leq 1000$, $K \leq 1000$
6	8	$K \leq 100$
7	16	$N \leq 200$
8	23	No additional constraints.

Example

Consider the following call:

```
path_sum(4, 5, 7, [0, 1, 2, 3, 0], [1, 3, 1, 1, 0])
```

There are 4 cities and 5 roads.

- Road 0 connects cities 0 and 1.
- Road 1 connects cities 1 and 3.

- Road 2 connects cities 2 and 1.
- Road 3 connects cities 3 and 1.
- Road 4 connects city 0 to itself.

The first 7 walks, written as road-index sequences, are: $[0, 1]$, $[0, 3]$, $[4, 0, 1]$, $[4, 0, 3]$, $[0, 0, 0, 1]$, $[0, 0, 0, 3]$, and $[0, 1, 1, 1]$. Their lengths are $2, 2, 3, 3, 4, 4, 4$, the sum of which is 22.

The procedure should therefore return 22.

Sample Grader

Input format:

```
N M K
U[0] V[0]
U[1] V[1]
...
U[M-1] V[M-1]
```

Output format:

```
R
```

Here, R is the value returned by `path_sum`.