January 2023 CSE106

Online Assignment on Graph Traversal

Section: B1 + B2 Marks: 10 Time: 30 Minutes

In task-3, we ran BFS multiple times with a complexity of O(n*m*q).

Suppose, our friend Alice has a superpower who can predict the final square where all the riders will gather with 99.99% accuracy. But he is pretty weak in adding integers, so he can't tell the total minimal number of moves correctly.

Now it is your duty to count the total number of moves, given a square to gather all the riders.

You must do this in a single BFS i.e. complexity of O(n*m).

Solving in **multiple** BFS will carry only a 40% mark.

Input:
The first line of input contains three integers n, m and q (1≤n≤100,1≤m≤100, 1≤q≤n*m) — the number of rows, the number of columns and the number of k-rider on the chessboard.

The following q line each has **three** integers x, y and k — (x,y) the coordinates of the rider, k is the number of maximum jumps this k-rider can make in a single move.

The final line of input contains two integers x and y — (x,y) the coordinates of the final square.

Here (x,y) are 0-indexed. Please refer to the sample IO for a better understanding of the input.

Output:
Output the total minimal number of moves, if we can move all the k-riders to the mentioned square. Else print -1.

Sample I/O:

Input	Output

3 4 2
0 0 1
0 1 3
2 3
	5

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

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

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