**Gridland Metro**

118.33 more points to get your next star!

Rank: **335341**|Points: **356.67/475**

Problem Solving

**You have successfully solved Gridland Metro**

You are now 118.33 points away from the 4th star for your problem solving badge.

[**Try the next challenge**](https://www.hackerrank.com/challenges/icecream-parlor?isFullScreen=false) | [**Try a Random Challenge**](https://www.hackerrank.com/challenges/maximum-subarray-sum)

* [**Problem**](https://www.hackerrank.com/challenges/gridland-metro/problem)
* [Submissions](https://www.hackerrank.com/challenges/gridland-metro/submissions)
* [Leaderboard](https://www.hackerrank.com/challenges/gridland-metro/leaderboard)
* [Discussions](https://www.hackerrank.com/challenges/gridland-metro/forum)
* [Editorial](https://www.hackerrank.com/challenges/gridland-metro/editorial)

The city of Gridland is represented as an  matrix where the rows are numbered from  to  and the columns are numbered from  to .

Gridland has a network of train tracks that always run in straight horizontal lines along a row. In other words, the start and end points of a train track are  and , where  represents the row number,  represents the starting column, and  represents the ending column of the train track.

The mayor of Gridland is surveying the city to determine the number of locations where lampposts can be placed. A lamppost can be placed in any cell that is *not occupied* by a train track.

Given a map of Gridland and its  train tracks, find and print the number of cells where the mayor can place lampposts.

**Note:** A train track may overlap other train tracks within the same row.

For example, if Gridland's data is the following:

k = 3

r c1 c2

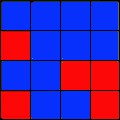
1 1 4

2 2 4

3 1 2

4 2 3

It yields the following map:



In this case, there are five open cells (red) where lampposts can be placed.

**Function Description**

Complete the *gridlandMetro* function in the editor below. It should return an integer that denotes the number of cells where lampposts can be installed.

gridlandMetro has the following parameter(s):

* *n*: an integer, the number of rows in Gridland
* *m*: an integer, the number of columns in Gridland
* *k*: an integer, the number of tracks
* *track*: a 2 dimensional array of integers where each element contains  integers that represent

**Input Format**

The first line contains three space-separated integers  and , the number of rows, columns and tracks to be mapped.

Each of the next  lines contains three space-separated integers,  and , the row number and the track column start and end.

**Constraints**

**Output Format**

Print a single integer denoting the number of cells where the mayor can install lampposts.

**Sample Input**

4 4 3

2 2 3

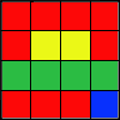
3 1 4

4 4 4

**Sample Output**

9

**Explanation**



In the diagram above, the yellow cells denote the first train track, green denotes the second, and blue denotes the third. Lampposts can be placed in any of the nine red cells.

#include<bits/stdc++.h>

#define pb push\_back

#define INF 1e18

#define vec vector<int>

#define ll long long

#define REP(i,a,b) for(i=a;i<b;i++)

using namespace std;

int main()

{

  ios\_base::sync\_with\_stdio(false);

  cin.tie(NULL);

  cout.tie(NULL);

  ll int n,m,k;

  cin>>n>>m>>k;

  map<ll int,vector<pair<ll int,ll int>>> p;

  for(ll int i=0;i<k;i++)

  {

    ll int r,c1,c2;

    cin>>r>>c1>>c2;

    if(c1>c2)

        swap(c1,c2);

    p[r].pb({c1,c2});

  }

  ll int ans=n\*m;

  for(auto it: p)

  {

    int x=0;

    sort(p[it.first].begin(),p[it.first].end());

    for(auto ele: p[it.first])

    {

      if(ele.first>x)

        ans-=ele.second-ele.first+1,x=ele.second;

      else if(ele.second>x)

        ans-=ele.second-x,x=ele.second;

    }

  }

  cout<<ans<<"\n";

}