<https://www.hackerrank.com/contests/world-codesprint-7/challenges/gridland-metro>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace ConsoleApplication1

{

class Program

{

static List<string> intercalarIntervalos(string a, string b)

{

string[] a\_split = a.Split(' ');

string [] b\_split = b.Split(' ');

int a\_ini = int.Parse(a\_split[0]);

int a\_fin = int.Parse(a\_split[1]);

int b\_ini = int.Parse(b\_split[0]);

int b\_fin = int.Parse(b\_split[1]);

List<string> intervalos = new List<string> ();

if (a\_ini < b\_fin && b\_ini <= a\_fin)

{

//intervalos.Add(a\_ini + " " + b\_fin);

intervalos.Add(Math.Min(a\_ini, b\_ini) + " " + Math.Max(a\_fin, b\_fin));

}

else if (b\_ini < a\_fin && b\_fin >= a\_ini)

{

//intervalos.Add(b\_ini + " " + a\_fin);

intervalos.Add(Math.Min(b\_ini, a\_ini) + " " + Math.Max(b\_fin, a\_fin));

}

return intervalos;

}

static void Main(string[] args)

{

//-------------------------

//List<string> lista = new List<string>();

//using (StreamReader esc = new StreamReader("C:\\test.txt"))

//{

// string line;

// while ((line = esc.ReadLine()) != null)

// {

// //System.Console.WriteLine(line);

// lista.Add(line);

// }

//}

//int indice\_lista = 0;

//-------------------------

//string[] input1 = "402159386 855281517 951".Split(' ');

string[] input1 = Console.ReadLine().Split(' ');

long n = long.Parse(input1[0]);

long m = long.Parse(input1[1]);

long k = long.Parse(input1[2]);

//HashSet<string> filas = new HashSet<string>();

Dictionary<long, List<string>> filas = new Dictionary<long, List<string>>();

for (int i = 0; i < k; i++)

{

string[] input2 = Console.ReadLine().Split(' ');

long r = long.Parse(input2[0]);

long c1 = long.Parse(input2[1]);

long c2 = long.Parse(input2[2]);

string intervalo\_actual = c1.ToString() + " " + c2.ToString();

if (filas.ContainsKey(r))

{

List<string> value = filas[r];

foreach (string inter in value)

{

List<string> mezclados =

intercalarIntervalos(intervalo\_actual, inter);

if (mezclados.Count == 1)

{

value.Remove(intervalo\_actual);

value.Remove(inter);

List<string> nueva\_lista = new List<string>();

nueva\_lista.AddRange(value);

nueva\_lista.Add(mezclados[0]);

filas[r] = nueva\_lista;

break;

}

else if (mezclados.Count == 0)

{

filas[r].Add(intervalo\_actual);

break;

}

}

}

else

{

List<string> aux = new List<string>();

aux.Add(intervalo\_actual);

filas[r] = aux;

}

}

long train\_track = 0;

foreach (KeyValuePair<long, List<string>> kvp in filas)

{

foreach (string elem in kvp.Value)

{

string[] inter = elem.Split(' ');

train\_track += long.Parse(inter[1]) - long.Parse(inter[0])+1;

}

}

Console.WriteLine((n \* m) - train\_track);

Console.ReadLine();

}

}

}