Chegg Study Textbook Solutions Expert Q&A Study Pack Practice

Find solutions for your homework Search

home / study / engineering / computer science / computer science questions and answers / using c++, data structures, c++ stl, all test inputs and o...

Question: Using C++, data structures, C++ STL, all test inputs and output...

Using C++, data structures, C++ STL, all test inputs and outputs are shown below.

Reunion

Description

It's been years since Kyaru and her best friends graduated from high school and they all miss each other very much. Recently, they've finally find a time when everyone is free. However, where to have a reunion becomes another problem to them. The reason is that they are currently living in different cities and plan to reunite in one city by plane, but there aren't many airlines in the whole country, which means there may be some unreachable cities for some of Kyaru's friends no matter how they fly.

More specifically, there're K friends (including Kyaru herself) living in N cities. In the whole country, there're M airlines, each of which is one-way and has one departure city and one destination city.

Of course an airline won't have its destination the same as its departure city. Also, a city is definitely reachable to whom lives there.

Now Kyaru would like you to help her find how many cities can be chose for reunion, i.e. reachable to all of them(You don't need to worry about how they go back after the reunion).

Input

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

The i-th line of the next K line contains a single integer $(1..N)$ representing the index number of the city where the i-th friend currently lives.

The i-th line of the next M line contains two space-separated integers a, b representing that there's an one-way airline from city a to city b.

Output

One integer indicating the number of cities that can be chose for reunion.

Sample Input/Output

Input

```
2 4 4
2
3
1 2
1 4
2 3
3 4
```

Output

```
2
```

Hint

They can reunite in city 3 or city 4

Constraint

$1 \leq K \leq 100, 1 \leq N \leq 1000, 1 \leq M \leq 10^4$.

Show transcribed image text

View comments (1) >

Post a question

Answers from our experts for your tough homework questions

Enter question

Continue to post

20 questions remaining

Snap a photo from your phone to post a question

We'll send you a one-time dow link

888-888-8888 Text r

By providing your phone number, you agree to receive a automated text message with a link to get the app. Stan messaging rates may apply.

My Textbook Solutions



Principles... Fundament... Fundan

5th Edition 1st Edition 5th Editi

View all solutions

Chegg Study Textbook Solutions Expert Q&A Study Pack Practice

Anonymous answered this
136 answers

Was this answer helpful? 0 0

Coded in c++.

Programming comments are written.

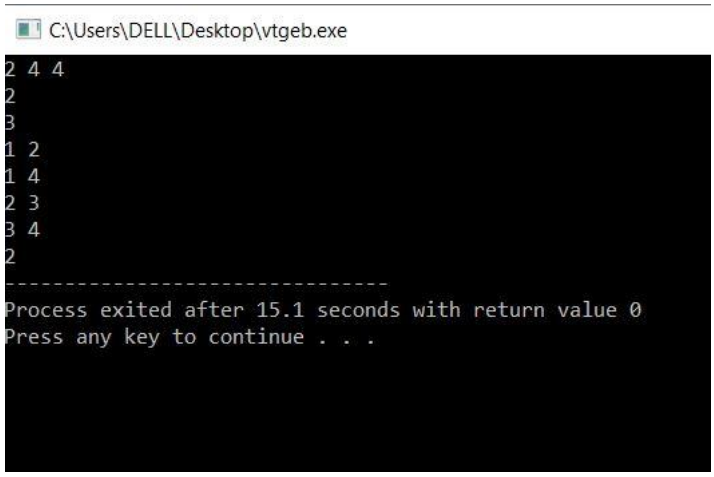
BFS is used to check if any friend can reach the city or not.

Iterated over each location if it is possible for reunion or not.

```
#include <iostream>
#include <vector>
#include <queue>
using namespace std;

//will return true only if we can go from start to end according to the ways
bool isPathBFS(vector <vector<int>>& ways, int nodes, int start, int end){

   bool visited[nodes];//the nodes which are already visited will not be visite again
   for(int i = 0;i<nodes;++i){
      visited[i] = false;//firstly no node is visited
   }

   queue <int> traversal;
   traversal.push(start);
   visited[start-1] = true;//start is visited so it is given true

   while(!traversal.empty()){

      start = traversal.front();
      traversal.pop();

      for(int i = 0;i < (int)ways[start-1].size();++i){
         if(ways[start-1][i]==end)
               return true;

         if(visited[ways[start-1][i]-1]==false){

            traversal.push(ways[start-1][i]);
            visited[ways[start-1][i]-1] = true;
         }
      }
   }

   return false;
}
int main()
{
      int n,k,m;
      cin>>k>>n>>m;
      int friends[1000];//will contain location of ith friend
      for(int i=0;i<k;i++)
      {
            cin>>friends[i];
      }
      int count=0;

      vector <vector<int>> ways;
   ways.resize(n);
      for(int i=0;i<m;i++)
      {
            int start;
            int end;
            cin>>start>>end;
            ways[start-1].push_back(end);//making the directed graph
      }


      for(int i=1;i<=n;i++)//we will check if ith location is possible for reuinion
      {
            bool possible=true;
            for(int j=0;j<k;j++)
            {

                  possible=isPathBFS(ways,n,friends[j],i);//is jth friend can reach ith location
                  if(friends[j]==i)
                  {
                        possible=true;
                  }
                  if(possible==false)//if any friend can't reach ith location we will stop array and look for
another reunion city
                  {
                        break;
                  }
```

```
            {
                count++;
            }
        }
        cout<<count;//will print number of places possible
}
```

Output window, here last entry is the actual output of the code and other are inputs:



View comments (2) >

Up next for you in Computer Science

/* * Instructions: * * 1. Only complete the functions specified below. Do not create any additional function. * 2. Use Visual Studio 2019 to build, test and

See answer

3.2. inpost.cpp This file should contain a definition for the following function: string convert(const string& infix) This function converts

See answer

See more questions for subjects you study

Chegg Study Textbook Solutions Expert Q&A Study Pack Practice