

Thomas' Shops

Overview

In this project you are going to implement a solution for a min-cut problem.

Problem

Mr. Thomas Jr. had always been a successful manager of East&West Co. He set up his own commerce network and has served the society with his many businesses in this network. However, he has become an ill-tempered old man in his eighties now, and he can not compete against e-commerce.

Therefore, he decided to close down some shops in his network and make it disconnected so that his imperial company becomes two split-ups: East Company and West Company. He does not want to do that, but he has to!

Find the minimum number of shops he should close down.

1. Shops are connected to each other by roads.
2. Two shops trade **if and only if** there is a road between them.
3. A shop in his network trade **only** with the shops in the network. Note that a shop does not trade with itself.
4. His network is **connected**, ie. there is always a path from any shop to any other shop.
5. Mr. Tom was a perfectionist, he invested great deal of care to make roads between shops **equal** to each other. A road is not wider than the other, or narrower, or better ... ie. Tom's network is unweighted.
6. If a shop is closed, all roads that pass through this shop get unavailable for trade. ie. These roads are **removed** from the network too.
7. He wants to, therefore disconnect his network in two, by closing down the **minimum** number of shops.

Write a program to find out the minimum number of shops.

Implementation

In your implementation you will read one file which is formatted as:

1. The very first line has two variables separated by a space: **numberOfShops** and **numberOfRoads** respectively. Those variables correspond to the number of shops and number of roads contained in Thomas' commerce network.
2. After the first line, there will be lines in the count of **numberOfRoads**. Each line will describe a road. For each those lines, you will have two **shopID** variables which are separated by a space. Those **shopIDs** are for what shops this road connects.

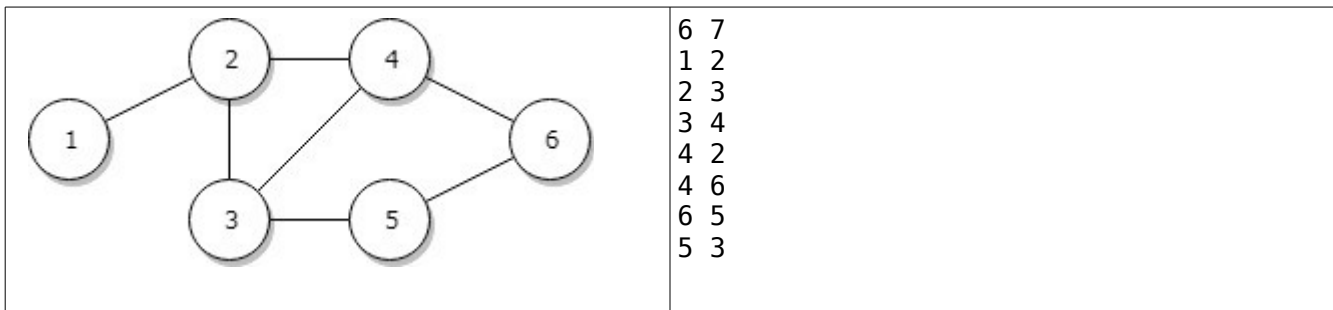
3. Constraints for the variables are as follows:

numberOfShops is an integer in range [2, 1000]

numberOfRoads is an integer in range [1, 1000]

shopID is an integer in range [1,1000]

4. An example file named **example.txt** is given below. Its content is at the right hand side, what network it depicts resides at the left hand side. It depicts a network with 6 shops and 7 roads within these shops.



5. The second line of **example.txt** defines a road between Shop 1 and Shop 2. You are guaranteed not to encounter duplicate road definitions. ie. If “1 2” exists, then there will not be “2 1” or “1 2” (again) through the end of the file.
6. There must always be trading shops in both East Co. and West Co. If it is not possible to split Tom's network this way, **print out 0**.
7. Use proper algorithms.
8. In case of any need, you are **only and only** allowed to include from Standard Template Library. Do not use external 3rd party libraries.
9. Try to follow an object-oriented methodology with well-chosen **variable**, **function**, **class names** and **comments** whenever possible.

Grading

Your project will be evaluated in terms of your implementation and your soft copy report.

Evaluation of the implementation (70 pts)

Your program should be compiled on ITU's Linux Server by the following command:

```
>g++ main.cpp -o thomassShops
```

```
>cat data.txt
6 7
1 2
2 3
3 4
4 2
4 6
6 5
5 3
>g++ main.cpp -o thomassShops
>./thomassShops data.txt
2
```

Please note that example output in red is **imaginary**.

Report of the analysis (30 pts)

1. Present your problem implementation **in detail**. (see **report.pdf** which is attached)

Policy

- You may discuss the problem addressed by the project at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your own, individual project.
- Academic dishonesty including but not limited to cheating, plagiarism, collaboration is unacceptable and subject to disciplinary actions.

Submission

- Please submit your project files through Ninova e-Learning System.
- You must submit a .zip file named **yourStudentId_p3.zip**. There will be two files in the zip. One is your implementation and one is your report.
 - all your source code in a single .cpp file named **main.cpp** You can define multiple classes in a single .cpp file if you need.
 - a soft-copy .pdf report named **report.pdf**
- All your implementation must be in C++, and we must be able to compile and run it on ITU's Linux Server (you can access it through SSH) using g++.
 - For Windows users: If you wish, you can use WinSCP to upload and edit your source code into ITU SSH Server, and use PuTTY to compile and run your algorithm. If does not, please make sure that your code is able to be compiled and runned on ITU's Linux Server.
- Your code **HAS TO** be compiled successfully. Otherwise you will get a grade of zero for the assignment, **even if you completed your report**.
- You should be aware that the Ninova e-Learning System clock may not be synchronized with your computer, watch, or cell phone. **Do not e-mail** the teaching assistant or the instructors about your submission after the Ninova site is closed for submission. If you have submitted to Ninova once and still want changes in your report, you should do this before the Ninova submission system closes. Your changes will not be accepted by e-mail. Connectivity problems in the last minute about the internet or about Ninova are not valid excuses for being unable to submit. You should not risk your project by leaving its submission to the last minute. After uploading to Ninova, make sure that your project appears there.
- **Start early**. If a question is not clear, e-mail the teaching assistant kcengiz@itu.edu.tr