H   🏠 Practice   🕐 Compete   💼 Jobs   🎖 Rank   🏆 Leaderboard

Network administration 🔖

🐝 by shaka_shadows

Problem	Submissions	Leaderboard	Discussions	Editorial 🔒

Time Limits C:5, Cpp:5, C#:6, Java:8, Php:18, Ruby:20, Python:20, Perl:18, Haskell:10, Scala:14, Javascript:20, Pascal:5

Like every IT company, the Uplink Corporation has its own network. But, unlike the rest of the companies around the world, Uplink's network is subject to very specific restrictions:

- Any pair of servers within the network should be directly connected by at most 1 link.

- Each link is controlled by some specific network administrator.

- No server has more than 2 links connected to it, that are controlled by the same administrator.

- For easier management, links controlled by some administrator cannot be redundant (this is, removing any link will disconnect some two previously connected servers)

Notice that 2 connected servers might not have any direct link between them. Furthermore, in order to keep the network in a secured status, Uplink directives periodically try to perform some modifications over the network to mislead hackers. The problem is, having such a huge network, they need a software to efficiently simulate the network status after any of such modifications. You have been assigned to write the core section of that software.

Operations performed by the directives are:

- Change the administrator assigned to some particular link.

- Place some number of security devices along a particular link.

Also, given a network administrator, they would like to know how many devices are in the path created by links controlled by that administrator (if any) between 2 servers.

Input Format
Input begins with a line containing 4 integers S, L, A, T separated by a single whitespace, denoting the number of servers, links, network administrators and transformations, respectively. L lines follow each one with 3 integers $x, y \ (x < y)$ and $a_i \ (1 \le a_i \le A)$, saying that there is a link between server x and server y, and that link is controlled by administrator a_i. Initially, network topology fulfills the restrictions described above and there is no security device along any link. Remaining T lines in the input follow one the next formats:

- $1 \ A \ B \ a_i$

meaning that link between server A and server $B \ (A < B)$ is requested to be assigned to administrator a_i

- $2 \ A \ B \ x$

meaning that the number of security devices along the link between server A and server $B \ (A < B)$ will be fixed to x, removing any existing devices on this link before the operation. The involved link will always exist.

- $3 \ A \ B \ a_i$

meaning that directives want to know the number of security devices placed along the path between server A and server B, just considering links controlled by administrator a_i.

Output Format
For each network transformation in the form $1 \ A \ B \ a_i$ you should output:

- "Wrong link" if there is no direct link between server A and server B.

- "Already controlled link" if the requested link does exist, but it is already controlled by administrator a_i.

- "Server overload" if administrator a_i already controls 2 links connected to one of the involved servers.

- "Network redundancy" if the requested assignment creates no new connection considering just the links controlled by a_i.

- "Assignment done" if none of the above conditions holds. In this case, link directly connecting A with B is assigned to a_i.

For each network transformation in the form $3\ A\ B\ a_i$ you should output:

- "No connection" if there is no path between the requested servers considering just the links controlled by a_i.

- "D security devices placed" where **D** is the number of security devices placed so far on the existing connection between the requested servers considering just the links controlled by a_i.

Constraints

$1 \le S \le 10^5$
$1 \le L \le 5 \times 10^5$
$1 \le A \le 10^2$
$1 \le T \le 5 \times 10^5$
$1 \le x \le 2000$

Sample Input:

```
4 5 3 15
1 2 1
2 3 1
3 4 2
1 4 2
1 3 3
2 3 4 49
1 1 2 3
2 1 4 64
3 1 4 2
1 1 2 3
3 4 2 3
3 1 3 3
1 1 4 3
3 3 4 2
3 2 4 1
2 1 4 13
2 1 3 21
2 2 3 24
1 2 3 3
1 2 4 3
```

Sample Output:

```
Assignment done
64 security devices placed
Already controlled link
No connection
0 security devices placed
Server overload
49 security devices placed
No connection
Network redundancy
Wrong link
```

☆ ☆ ☆ ☆ ☆

Current Buffer (saved locally, editable) ⎇ 🕒   Java 7   ⌄   ⤢   ⚙

```
 1 ▾ import java.io.*;
 2   import java.util.*;
 3   import java.text.*;
 4   import java.math.*;
 5   import java.util.regex.*;
 6
 7 ▾ public class Solution {
 8
 9 ▾     public static void main(String[] args) {
10 ▾         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11         }
12   }
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature