

The Scenario

To assist with social networking, a schmoozing assistant is required. The assistant will store information on known relations between people and, given two people, identify the chain of relations that link them – if any.

For example, given an acquaintance with Michael Lonnon, who works for Sun, how can we get an introduction to James Gosling? The chain between them might be something like:

Michael Lonnon works with Peter Scopes,
 Peter Scopes friends with Brendan Hodgson,
 Brendan Hodgson friends with Scott Nealy,
 Scott Nealy works with James Gosling.

There are a number of ways in which two people may be related. For the purposes of schmoozing, some are more preferable to others.

The key to the schmoozing assistant is an algorithm for finding the shortest chain of relations between two people.

Consider the following sample set of immediate relations:

((John, Tim), friendship), ((John, Jane), workmates),
 ((Michael, Mark), family), ((Tim, Jane), friendship),
 ((Tim, Mark), friendship), ((Jane, Michael), workmates)

Let's say a chain can be represented by a sequence of names $\langle \text{name}_1, \dots, \text{name}_n \rangle$.

Say John wishes to schmooze his way to Michael, an algorithm for finding a chain might operate as follows:

1. Start with an initial set S of chains consisting of the people immediately related to John i.e. $\{ \langle \text{Tim} \rangle, \langle \text{Jane} \rangle \}$.
2. While S does not contain a chain to Michael do the following:
 - a. Remove a chain c from S e.g. $\langle \text{Tim} \rangle$.
 - b. Find the people immediately related to the last name in the chain e.g. John, Jane, Mark.
 - c. For each name n , create a new chain c' by appending n to c , and then add it to S e.g. add $\langle \text{Tim}, \text{John} \rangle$, $\langle \text{Tim}, \text{Jane} \rangle$, and $\langle \text{Tim}, \text{Mark} \rangle$ to $\{ \langle \text{Jane} \rangle \}$.

Note that there are a number of issues that need to be dealt with in a complete solution. For instance, "cycles" chains. If a chain contains a cycle such as, in terms of the example, $\langle \text{Tim}, \text{John} \rangle$, or $\langle \text{Tim}, \text{Jane}, \text{Tim} \rangle$ then it should be discarded.

Your Challenge

Your challenge is to construct a skeleton schmoozing assistant that will store or have access to data on the immediate relations between people and will provide a method for finding a chain that is based on the given algorithm, or on one of your own design.

You may need classes to represent:

- People
- Relations
- Chains

A basic method for finding a chain will simply return a sequence of names; a better method will return details of names and relations.

You may wish to use the example data in a test harness to ensure your construction operates correctly.

Possible Extensions

One option is to add file loading so that relations and people can be loaded from a text file, where the first line is the name of the user and lines thereafter are comma separated values describing relationships. For example:

```
User name  
John Doe, Michael Hunt, friend  
...  
Jane Smith, Ben Durham, colleague
```

To extend this further, you may wish to add file saving so that updated relations can be saved.

Another option would be to add merging, so that two people could share data. You might need to consider who has priority if there are contradicting entries for the same relation.

However, if you desire to extend this challenge in another suitable way, feel free to do so.