You wrote a trendy new messaging app, MeshMessage, to get around flaky cell phone coverage.

Instead of routing texts through cell towers, your app sends messages via the phones of nearby users, passing each message along from one phone to the next until it reaches the intended recipient. (Don't worry—the messages are encrypted while they're in transit.)

Some friends have been using your service, and they're complaining that it takes a long time for messages to get delivered. After some preliminary debugging, you suspect messages might not be taking the most direct route from the sender to the recipient.

Given information about active users on the network, find the shortest route for a message from one user (the sender) to another (the recipient). Return a list of users that make up this route.

> There might be a *few* shortest delivery routes, all with the same length. For now, let's just return *any* shortest route.

Your network information takes the form of a dictionary mapping username strings to a list of other users nearby:

```python
network = {
    'Min'     : ['William', 'Jayden', 'Omar'],
    'William' : ['Min', 'Noam'],
    'Jayden'  : ['Min', 'Amelia', 'Ren', 'Noam'],
    'Ren'     : ['Jayden', 'Omar'],
    'Amelia'  : ['Jayden', 'Adam', 'Miguel'],
    'Adam'    : ['Amelia', 'Miguel', 'Sofia', 'Lucas'],
    'Miguel'  : ['Amelia', 'Adam', 'Liam', 'Nathan'],
    'Noam'    : ['Nathan', 'Jayden', 'William'],
    'Omar'    : ['Ren', 'Min', 'Scott'],
    ...
}
```

For the network above, a message from Jayden to Adam should have this route:

```python
['Jayden', 'Amelia', 'Adam']
```

Want more coding interview help?

Check out **interviewcake.com** for more advice, guides, and practice questions.