Project 11.3 The Oracle of Bacon

In the 1990's, a group of college students invented a pop culture game based on the six degrees of separation idea called the "Six Degrees of Kevin Bacon." In the game, players take turns challenging each other to identify a connected chain of actors between Kevin Bacon and another actor, where two actors are considered to be connected if they appeared in the same movie. An actor's Bacon number is the distance of the actor from Kevin Bacon.

This game later spawned the "Oracle of Bacon" website, at `http://oracleofbacon.org`, which gives the shortest chain between Mr. Bacon and any other actor you enter. This works by constructing a network of actors from the Internet Movie Database (IMDb)[7], and then using breadth-first search to find the desired path. This network of movie actors provides fertile ground for investigating all of the concepts discussed in this chapter. Is the network really a small-world network, as suggested by the "Six Degrees of Kevin Bacon?" Is there really anything special about Kevin Bacon in this network? Or is there a short path between any two pair of actors? Is the network scale-free?

In this project, you will create an actor network, based on data from the IMDb, and then answer questions like these. You will also create your own "Oracle of Bacon."

Part 1: Create the network

On the book website, you will find several files, created from IMDb database files, that have the following format:

```
League of Their Own, A (1992)▷Tom Hanks▷Madonna▷Penny Marshall▷ ···
Animal House (1978)▷Kevin Bacon▷John Belushi▷Donald Sutherland▷ ···
Apollo 13 (1995)▷Kevin Bacon▷Tom Hanks▷Bill Paxton▷Gary Sinise▷ ···
```

The files are tab-separated (the ▷ symbols represent tabs). The first entry on each line is the name of a movie. The movie is followed by a list of actors that appeared in that movie.

Write a function

```
createNetwork(filename)
```

that takes the name of one of these data files as a parameter, and returns an actor network (as an adjacency list) in which two actors are connected if they have appeared in the same movie. For example, for the very short file above, the network would look like that in Figure 1. Each link in this network is labeled with the name of a movie in which the two actors appeared. This is not necessary to determine a Bacon number, but it is necessary to display all the relationships, as required by the traditional "Six Degrees of Kevin Bacon" game (more on this later).

The following files were created from the IMDb data, and are available on the book website.

[7] `http://www.imdb.com`

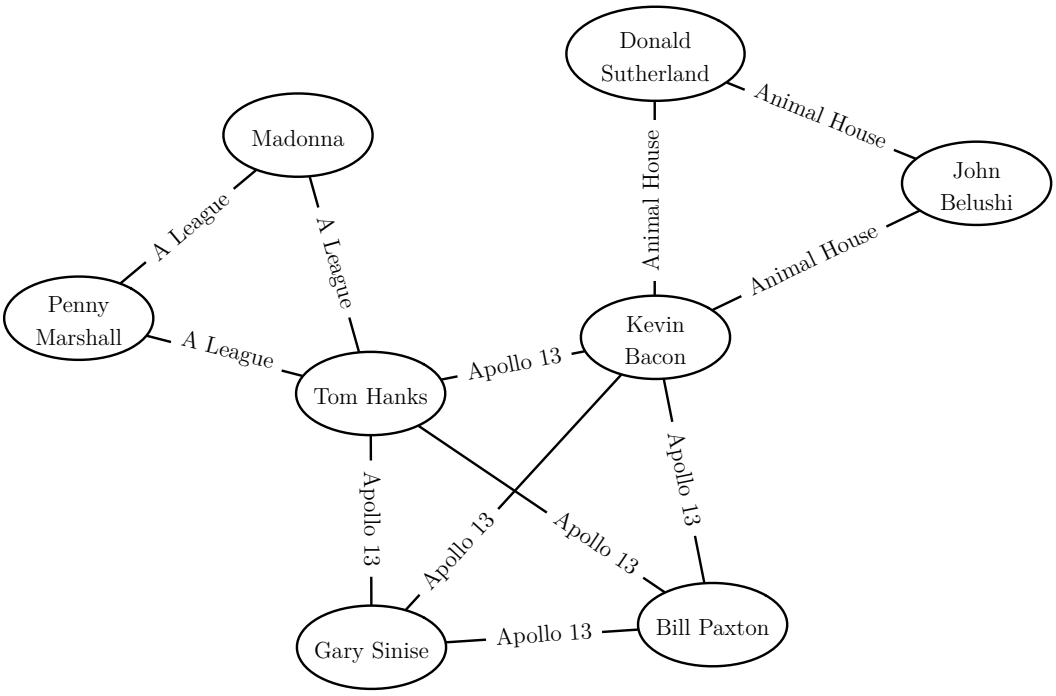

Figure 1 A simple actors network.

File	Movies	Actors	Description
`movies2005.txt`	560	24,276	MPAA-rated movies from 2005
`movies2012.txt`	518	25,225	MPAA-rated movies from 2012
`movies2013.txt`	517	24,190	MPAA-rated movies from 2013
`movies2000p.txt`	8,396	362,798	MPAA-rated movies from 2000–2014
`movies2005p.txt`	5,590	251,881	MPAA-rated movies from 2005–2014
`movies2010p.txt`	2,547	117,908	MPAA-rated movies from 2010–2014
`movies_mpaa.txt`	12,404	514,780	All movies rated by MPAA
`moves_all.txt`	565,509	6,147,773	All movies

Start by testing your function with the smaller files, building up to working with the entire database. Keep in mind that, when working with large networks, this and your other functions below may take a few minutes to execute. For each file, we give the number of movies, the total number of actors in all casts (not unique actors), and a description.

Part 2: Is the network scale-free?

Because the actor network is so large, it will take too much time to compute the average distance between its nodes and its clustering coefficient. But we can plot the degree distribution and investigate whether the network is scale-free. To do so, write a function

```
degreeDistribution(network)
```

that plots the degree distribution of a network using `matplotlib`. Again, start with small files and work your way up.

Question 11.3.1 *What does your plot show? Is the network scale-free?*

Question 11.3.2 *Where does Kevin Bacon fall on your plot? Is he a hub? If so, is he unique in being a hub?*

Question 11.3.3 *Which actors have the ten largest degrees? (They might not be who you expect.)*

Part 3: Oracle of Bacon

You now know everything you need to create your own "Oracle of Bacon." Write a function

```
oracle(network)
```

that repeatedly prompts for the names of two actors, and then prints the distance between them in the network. Your function should call `bfs` to find the shortest distance.

For an extra challenge, modify the algorithm (and your adjacency list) so that it also prints how the two actors are related. For example, querying the oracle with Kevin Bacon and Zooey Deschanel might print

```
Zooey Deschanel and Kevin Bacon have distance 2.

1. Kevin Bacon was in "The Air I Breathe (2007)" with Clark Gregg.
2. Clark Gregg was in "(500) Days of Summer (2009)" with Zooey Deschanel.
```

Part 4: Frequencies of Bacon numbers

Finally, determine just how central Kevin Bacon is in the movie universe. To do so, create a chart with the frequency of Bacon numbers. Write a function

```
baconNumbers(network)
```

that displays this chart for the given network. For example, given the file `movies2013.txt`, your function should print the following:

```
Bacon Number   Frequency
------------   ---------
           0           1
           1         152
           2        2816
           3       12243
           4        4230
           5         253
           6          47
           7           0
           8           0
    infinity         819
```

An actor has infinite Bacon number if he or she cannot be reached from Kevin Bacon.

Once your function is working, call it with some of the larger files.

Question 11.3.4 *What does the chart tell you about Kevin Bacon's place in the movie universe?*