

# **CIS 2168: Assignment #10**

Due on Monday, December 8, 2014

*11:59pm*

**Anwar Mamat**

This is a group project. Two heads are better than one. Two students can work together.

## Problem 1

### Six Degrees of Kevin Bacon

In this assignment, you write a program that will help you play the “Six Degrees of Kevin Bacon” game. This game is played on a graph of movies and actors/actresses who starred in them. The Bacon Number is defined as follows: Kevin Bacon has a Bacon Number of 0; people who co-starred in a movie with Kevin Bacon have a Bacon Number of 1; people who co-starred in a movie with someone who has a Bacon Number of 1 in turn have a Bacon Number of 2; and so on. Given any actor/actress, the goal of the game is to establish their Bacon Number by a sequence of movies and co-stars. As shown in Figure 1, Bill Murray has

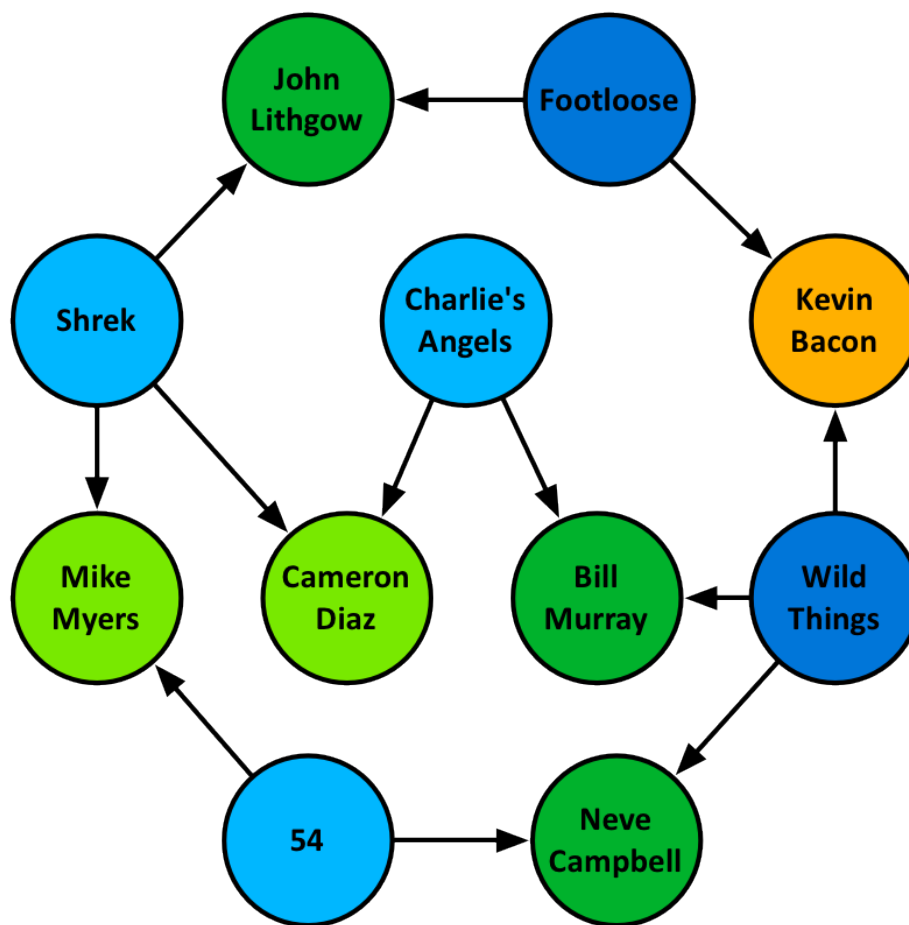


Figure 1: Six Degrees of Kevin Bacon

a Kevin Bacon number of 1, because co-starred in “Wild Things”, while Cameron Diaz has a Kevin Bacon Number of 2, because she she co-starred with Bill Murray in “Chailie’s Angels”.

For example, using the “action” data set given below, we can show that “Al Pacino” has a Bacon Number of 3 as follows:

Pacino, Al
------------

```

Heat (1995)
Rosales Jr., Thomas
Lost World: Jurassic Park, The (1997)
5 Richards, Ariana
Tremors (1990)
Bacon, Kevin

```

Kevin Bacon (0) starred in Tremors with Richards, Ariana (1) who starred in Lost World: Jurassic Park, The (1997) with Rosales Jr., Thomas (2) who starred in Heat (1995) with Pacino, Al (3). If we use the "all cast" data set given below, we can show that "Al Pacino" has a Bacon Number of 1 as follows:

```

Pacino, Al
Boffo! Tinseltown's Bombs and Blockbusters (2006)
Bacon, Kevin

```

Kevin Bacon (0) starred in Boffo! Tinseltown's Bombs and Blockbusters (2006) with Pacino, Al (1). Interestingly this works way in the past as well:

```

De Rosselli, Rex
Lion's Claws, The (1918)
Brinley, Charles
Adventure in Sahara (1938)
5 Lawrence, Marc (I)
Big Easy, The (1987)
Goodman, John (I)
Death Sentence (2007)
Bacon, Kevin

```

Turns out that Rex De Rosselli died in 1941 and yet he has a Bacon Number of just 4! Actually Rex really has a Bacon Number of 3 but we need to use a much bigger data set "cast.all.txt" to show this:

```

De Rosselli, Rex
Dangerous Adventure, A (1922)
McCullough, Philo
Chamber of Horrors (1966)
5 Danova, Cesare
Animal House (1978)
Bacon, Kevin

```

We have included two data sets suitable for the program in the archive above (the data sets are courtesy of Robert Sedgewick):

```

action06.txt (4.4 MB, only action movies)
cast.all.txt (64 MB, all movies from the 2014 IMDB)

```

The format of these data sets is rather simple: Each line is a movie, and each movie consists of several fields separated by the "/" character. The first field is the name of the movie itself, all the following fields are the names of actors and actresses starring in the movie. For example:

```

Heat (1995)/Daniels, Max/Perry, Manny/Marzan, Rick/Pacino, Al, ...
Heat After Dark (1996)/Kitami, Toshiyuki/Sugata,...
Heated Vengeance (1985)/Dye, Cameron/Walker, Robert (III), ...

```

Reading this data is not complicated, but we hand you the parsing code anyway so you can focus on the search algorithm instead.

In order to find the smallest Bacon Number for an actor we proceed as follows: First we identify the vertices for both Kevin Bacon and the actor in question (you already have that code). Then we start a breadth-first search at the vertex for Kevin Bacon; as we do this we keep track of the "previous vertex" that got us to the one we're currently investigating. Once we find the vertex for the other actor, we are done: We just have to print out the path that got us here. This implementation of BFS is the only thing you have to write for this problem!

You will implement following functions in the given NetBeans project:

```
1.Degree of separation from Kevin Bacon:
2.Degree of separation between any two actors/actresses:
3.Search actor/actress/movie:
4.List cast of a movie or movies of an actor/actress:
5. Exit
Select:
```

Here is what each menu item does:

- 1: Finds the Kevin Bacon number of an actor/actress
- 2: Find the shortest distance between any two actors/actresses.
- 3: Actors with same name has a Roman number after their names. You can enter the name and search the exact name in the database. For example, Emma Watson's name appeared as "Watson, Emma (II)" in the movie database.

```
Select:3
Enter the name:Watson, Emma
Watson, Emma (II)
```

- 4: If input is a movies, it lists all the case. If the input is an actor/actress, it lists all the movies he/she starred in. For example:

```
Select:4
Enter the name:Shawshank Redemption, The (1994)
Ciccolella, Jude
Thomas, Rohn
...
Freeman, Morgan (I)
...
Robbins, Tim (I)
...
Wallace, Cornell
```

Tom Hanks movies:

```
Select:4
Enter the name:Hanks, Tom
burbs, The (1989)
Angels & Demons (2008)
5 Apollo 13 (1995)
Bachelor Party (1984)
Big (1988)
Boffo! Tinseltown's Bombs and Blockbusters (2006)
10 Bonfire of the Vanities, The (1990)
```

```
10 Cars (2006)
    Cast Away (2000)
    Catch Me If You Can (2002)
    Celluloid Closet, The (1995)
    Charlie Wilson's War (2007)
15 Cold Case, A (2008)
    Da Vinci Code, The (2006)
    Dragnet (1987)
    Elvis Has Left the Building (2004)
    Every Time We Say Goodbye (1986)
20 Forrest Gump (1994)
    Great Buck Howard, The (2007)
    Green Mile, The (1999)
    He Knows You're Alone (1980)
    How Starbucks Saved My Life (2008)
25 Joe Versus the Volcano (1990)
    Ladykillers, The (2004)
    League of Their Own, A (1992)
    Magnificent Desolation: Walking on the Moon 3D (2005)
    Man with One Red Shoe, The (1985)
30 Money Pit, The (1986)
    Nothing in Common (1986)
    Philadelphia (1993)
    Polar Express, The (2004)
    Punchline (1988)
35 Radio Flyer (1992)
    Return with Honor (1998)
    Risk Pool, The (2008)
    Road to Perdition (2002)
    Saving Private Ryan (1998)
40 Sleepless in Seattle (1993)
    Splash (1984)
    Terminal, The (2004)
    That Thing You Do! (1996)
    Toy Story (1995)
45 Toy Story 2 (1999)
    Turner & Hooch (1989)
    Volunteers (1985)
    Who Killed the Electric Car? (2006)
    You've Got Mail (1998)
50 Boone's Lick (2008)
    Talk of the Town (2009)
    Toy Story 3 (2009)
```

- 5: terminate the program.

The “Graph” class, “Bag” class are fully implemented for you. The Graph class represents an undirected graph of vertices named 0 through  $V - 1$ . It supports the following two primary operations: add an edge to the graph, iterate over all of the vertices adjacent to a vertex. It also provides methods for returning the number of vertices  $V$  and the number of edges  $E$ . Parallel edges and self-loops are permitted. This implementation uses an adjacency-lists representation, which is a vertex-indexed array of Bag objects. All operations take constant time (in the worst case) except iterating over the vertices adjacent to a given vertex, which takes time proportional to the number of such vertices. For additional help, you can refer to

the “<http://algs4.cs.princeton.edu/code/javadoc/Graph.html>”.

```
Graph g = new Graph(6);  
g.addEdge(0, 3);  
g.addEdge(0, 4);  
g.addEdge(1, 3);  
5 g.addEdge(1, 5);  
g.addEdge(4, 2);  
g.addEdge(4, 5);
```

represents this graph:

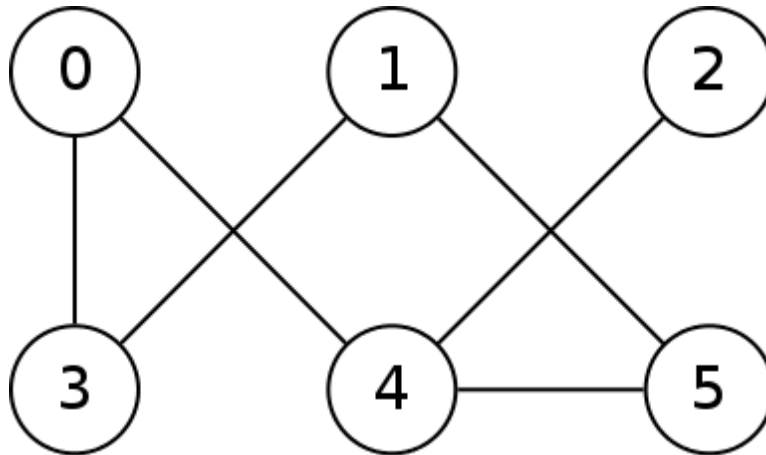


Figure 2: Undirected graph

You will have to add the mapping relationships between vertex symbols (name of a movie, actor, actress) and vertex number in “SymbolGraph” class. For example: in Graph 2, vertex 0 and vertex 5 can represent movies and other vertices represent actors/actresses. You also have to implement methods to find the shortest path between two given vertices. All edges have the weight of 1. Therefore, you can also use BFS to find the shortest distance between two vertices.

Before you are sure your program is running correctly, test your code with a smaller database base. Start with a file that only has 5-10 movies and actors/actresses. If you think your program is working, then try with “action06.txt”, (4.4 MB, only action movies). To try “cast.all.txt”, 64 MB, all movies from the 2014 IMDB, 300,000 movies, 900,000 actors/actresses, you have to allocate larger memory for your NetBeans. Google it for instructions to do that. If you run your program from command line you specify memory size in your command line. Following command allocate 2gb memory for JVM to run KevinBacon.

```
java -Xmx2g KevinBacon
```

## Grading

This is a **two-student group project**. You can find a partner from your class. If you cannot find a partner, or do not want to work with a partner, you can complete the assignment individually.

Homework is 100 points. 80 will reflect functionality and correctness. 20 points on your program will reflect your programming style, documentation. **If your code does not compile, you will not receive any credit.**

## Commenting and Documenting Code

Code must be properly commented. The main idea is that the grader should be able to understand your code easily, not have to tear his or her hair out wondering what some statement is doing. The first time you have to deal with poorly commented code (if you haven't already), you will understand how annoying it is. In particular, the top of each code file should contain your name, the course and assignment numbers, and a brief summary of what's in the file. Line-by-line comments should be included as necessary to make the code easy to read. A clear coding style, together with informative variable and function names, will reduce the number of comments required. Obscure code or cryptic function names will cause loss of points (for bad style) and also require more extensive comments.

## What to submit

A single zip file called Assignment10\_firstname\_lastname.zip, where firstname is your first name, and lastname is your last name. In this zip file, put:

1. Names of the group members
2. Java source
3. A README file with:
  - Instructions to compile and run of your code (include a description of command line options).
  - If your solution is not perfect, explain what parts you did and what part you did not do.
  - List of files submitted
  - All your data and results (in plain text files).
  - Anything else you want TA know
4. Submit this zip file to Blackboard