

# Graph Representations & Six Degrees of Separation (Extra Credit)

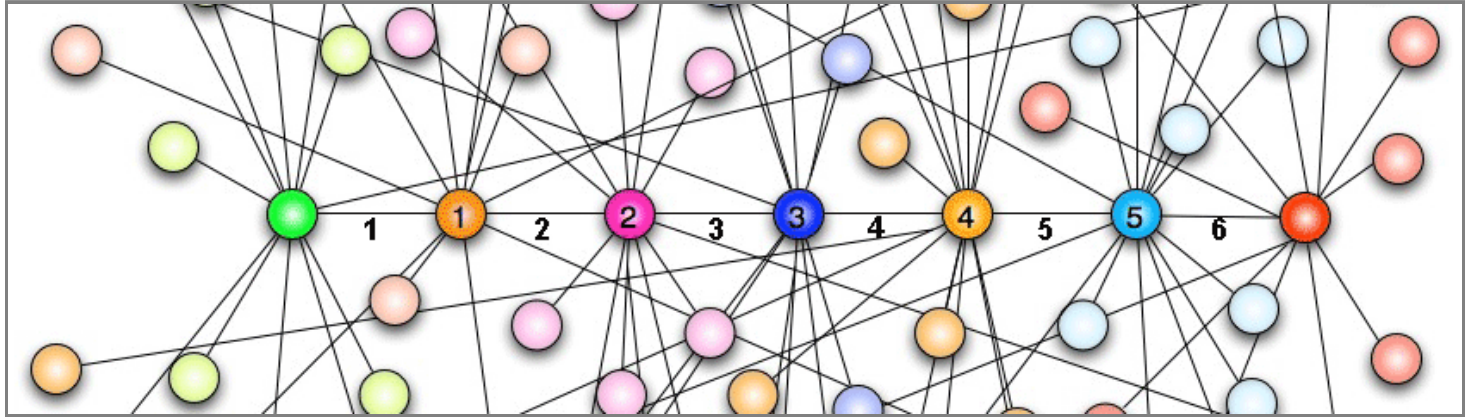
Submit Assignment

**Due** Jun 16, 2017 by 11:59pm

**Points** 0

**Submitting** a file upload

**File Types** zip



*Note: This assignment is extra-credit and is worth 100 points, weighted according to your assignment grade (30% of your final grade).*

## Overview

The number of relationships that exist between people in any problem domain is often surprising. Our world is becoming more and more interconnected, a concept that was explored by the social psychologist Stanley Milgram in the 1960s. Milgram is known for his "small world" experiment, which is described on the following wikipedia page: [https://en.wikipedia.org/wiki/Small-world\\_experiment](https://en.wikipedia.org/wiki/Small-world_experiment) ([https://en.wikipedia.org/wiki/Small-world\\_experiment](https://en.wikipedia.org/wiki/Small-world_experiment)).

## The Milgram Experiment

The goal of the experiment was to create a chain of correspondence between distant individuals who did not share a direct relationship. If the source of the correspondence did not know the destination individual, then perhaps it is likely they knew someone who knew the individual. As part of the experiment any individual who received a piece of correspondence was asked to:

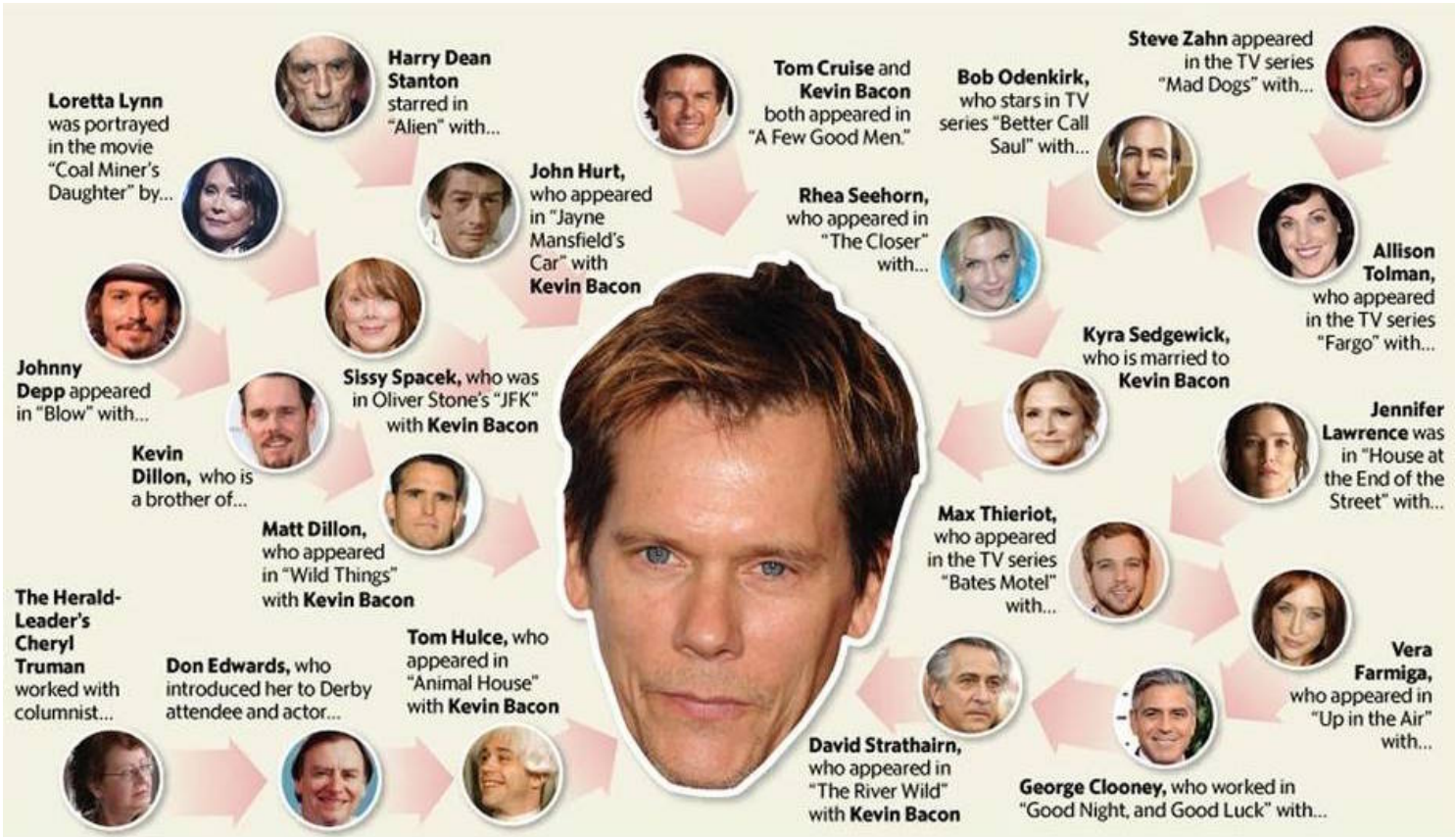
- forward the correspondence to the destination if they knew the individual
- otherwise forward the correspondence to relations that were likely to know the individual

## Six Degrees of Separation

The results of Milgram's experiment were surprising in that the correspondence often reached the destination in around 5-6 steps (or relationships). This resulted in the now-popular description of "six-degrees of separation."

## The Kevin Bacon Game

Initially created as a parlor game, the "Kevin Bacon Game" tried to link all actors/actresses in Hollywood with Kevin Bacon in less than six degrees. The game tracks actors/actresses who starred in the same movie and then provides a bacon number, which represents the minimum number of links that separate an actor/actress from Kevin Bacon. An example of these types of relationships can be seen below. Please note that the image below does not show the minimum number of links, but is an example of what types of links exist.



(image by Chris Ware cware@herald-leader.com)

This game clearly shows the concept of six-degrees of separation as it is difficult to find individuals with a high bacon number. For example, at the time of this writing the following bacon numbers can be found on google:

| Individual      | Bacon Number |
|-----------------|--------------|
| Justin Bieber   | 2            |
| Barrack Obama   | 2            |
| Ben Affleck     | 1            |
| Mark Zuckerberg | 3            |

Graph Theory

The phenomena of six-degrees of separation can be seen in all types of networks and is an interesting application of graph structures. The goal of this assignment is to store actors and movies in a graph and to calculate "Bacon numbers," with a twist. Instead of bacon numbers we will be writing a program that calculates the distance between actors/actresses and any central figure in Hollywood. For example, you could calculate "Lawrence numbers" for Jennifer Lawrence or "Connery numbers" for Sean Connery.

A question we want to directly answer is which actor/actress is really at the "center of Hollywood?"

## Setup

Download the following zip file: [link](#). This file contains several files with a subset of data for Hollywood actors and actresses. The file name tells you how many actors/actresses are in the file as well as the maximum number of movies tracked for that individual. Please note that each file is a random snapshot of the data available for that actor/actress. For many of these actors/actresses there are several hundred movies available from various data sets.

Each line of a file has the following format. Note that tv shows are included in the data set.

```
Actor/actress name | movie/tv show #1 | movie/tv show #2 | ...
```

Here is a few lines from one of the files:

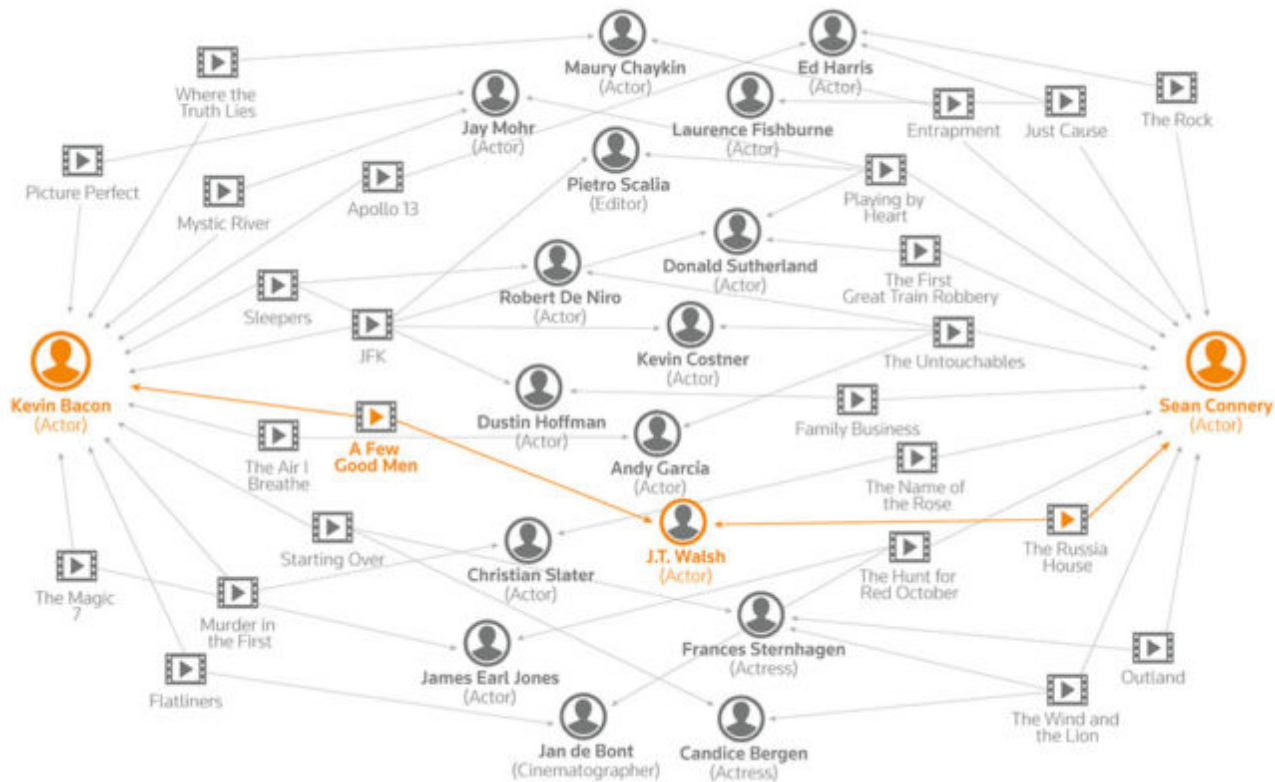
```
Astaire, Fred | Puttin' on His Top Hat | "The Dick Cavett Show" | ...
Baio, Scott | Battle of the Network Stars XIV | Circus of the Stars #5 | ...
Bale, Christian | Larger Than Life Adversaries | "The Drunken Peasants" | ...
Brosnan, Pierce | The Tailor of Panama | 25th Film Independent Spirit Awards
| ...
```

## Building a Graph

Your first programming task is to build a graph structure that can be used to generate "actor numbers." An actor number is like a "Bacon number" except that you can choose any actor as the source vertex. Your graph should have the follow properties:

- Edges are undirected
- Both actors and movies are represented in the graph by vertices
  - i.e. A vertex cannot contain an actor or a movie
- You can use an adjacency list or adjacency matrix, whichever your prefer

Write a routine that parses through an input file, from the zip file above and builds a graph using the data from the file. A visualization of what your graph might look like is given below:



(image by IMDb & Thomson Reuters Data Fusion)

## Calculating Actor Numbers

Create a new routine that accepts an actor name and calculates the "actor number" for all other actors in the graph above. For example:

```
public Map<String, Integer> generateActorNumbers(String actorSource)
{
    //do some work and return a map of actor to "actor numbers"
}
```

To calculate "actor numbers" you should use the breadth-first search on your graph (BFS) rather than depth-first search (DFS). You should take a moment and consider why this is the case. There are several complications when writing BFS here:

- You need to keep a counter for each set of neighbors traversed from your source vertex. For example:
  - All immediate neighbors should have an "actor number" of 1
  - All neighbors of immediate neighbors should have an "actor number" of 2
- You must skip all vertices that store movies as part of your calculation

## Console Program and Results

Write a console program that allows a user to do the following:

- Load an input file of their choice
- Builds a graph based on the input file
- Enter a source actor/actress name
- Lists interesting details for all actor numbers based on the source actor

Some example output is given below:

Welcome to the Six-Degrees of Separation Game!

Please choose one of the following files:

1. 20\_actors\_100\_movies.txt
2. 20\_actors\_15\_movies.txt
3. 300\_actors\_15\_movies.txt
4. 300\_actors\_5\_movies.txt
5. 300\_actors\_50\_movies.txt
6. 50\_actors\_5\_movies.txt
7. 900\_actors\_100\_movies.txt

Building graph, please wait...

Enter a source actor: Ball, Lucille

Ball, Lucille: 0

Bergen, Candice: 1

Cagney, James: 1

Carter, Lynda: 1

Chabert, Lacey: 1

Connery, Sean: 1

...

Banks, Tyra: 2

Bauchau, Patrick: 2

Bleeth, Yasmine: 2

Brando, Marlon: 2

Braugher, Andre: 2

Buscemi, Steve: 2

...

Arkin, Alan: 3

Boxleitner, Bruce: 3

Elster, Kevin: 3

Landau, Juliet: 3

Parker, Leni: 3

Scacchi, Greta: 3

Trejo, Danny: 3



Watson, Alberta: 3

Astaire, Fred: 4

The average "Actor Number" above is: 1.91

## Submission

- Submit your entire project to the Canvas drop-box zipped.
- Remember to follow our style guide and include Javadocs for your classes!

### Graph Representations & Six Degrees of Separation Rubric

| Criteria   | Ratings                   |                        | Pts      |
|--|---------------------------|------------------------|----------|
| Movie and actor data are correctly read/parsed from the given files.   | 10.0 pts<br>Full<br>Marks | 0.0 pts<br>No<br>Marks | 10.0 pts |
| A graph structure is built with undirected edges and vertices that can store either an actor or movie. An adjacency list or adjacency matrix is used.  | 15.0 pts<br>Full<br>Marks | 0.0 pts<br>No<br>Marks | 15.0 pts |
| Breadth-first-search is used to find actor numbers in your graph.  | 15.0 pts<br>Full<br>Marks | 0.0 pts<br>No<br>Marks | 15.0 pts |
| generateActorNumbers() correctly reports actor numbers for the given input. Only actors are reported with the results from the function (i.e. no movies are reported to have actor numbers). | 15.0 pts<br>Full<br>Marks | 0.0 pts<br>No<br>Marks | 15.0 pts |
| Console program allows a user to use any of the test files included with the assignment description.   | 10.0 pts<br>Full<br>Marks | 0.0 pts<br>No<br>Marks | 10.0 pts |
| Console program allows the user to enter an actor and actor numbers are printed using the format described above. The average actor number is also displayed.                                | 10.0 pts<br>Full<br>Marks | 0.0 pts<br>No<br>Marks | 10.0 pts |
| Styles: naming conventions, brace placement, spacing, commented code, redundancy, packages and magic numbers.  | 15.0 pts<br>Full<br>Marks | 0.0 pts<br>No<br>Marks | 15.0 pts |
| Formal documentation: full Javadocs & comment block header.  | 10.0 pts<br>Full<br>Marks | 0.0 pts<br>No<br>Marks | 10.0 pts |
| Total Points: 100.0  |                           |                        |          |