CSC 300 Data Structures – Fall 2015

Programming Assignment #4 – Graph Problems

Degrees of separation: One of the classic applications of shortest-paths algorithms is to find the *degrees of separation* of individuals in social networks. A recent form of this has been popularized by a game known as the *Kevin Bacon Game* or Six Degrees of Kevin Bacon. As you may know, Kevin Bacon is a Hollywood actor who has enjoyed a long career and starred in many movies. The saying goes that any other actor or actress in Hollywood can trace a path from themselves to Kevin Bacon in less than six degrees of separation.

Every performer is assigned a Bacon number which is calculated as follows: Bacon himself is 0, any performer who has been in the same cast as Bacon has a Bacon number of 1, any other performer (except Bacon) who has been in the same cast as a performer whose number is 1 has a Kevin Bacon number of 2, and so forth. Here are some examples to make the rules clear:

Kevin Bacon has never performed with Omar Sharif but he has a Bacon number of 2, because Sharif was in the 1979 film *Ashanti* with Winston Ntshona who appeared with Kevin Bacon in the 1994 movie, *The Air Up There*. Given the name of a performer, the simplest version of the game is to find an alternating sequence of movies and performers that lead to Kevin Bacon.

% Bacon_Number movie.txt

Enter the name of an actor or actress: Vlahos, Stephanie

Vlahos, Stephanie

Mavro gala (2000)

Souglakos, Apostolos

Escape to Athena (1979)

Gould, Elliot

Big Picture, The (1989)

Bacon, Kevin

Enter the name of an actor or actress: Dunham, Phil

Dunham, Phil

Singin' in the Rain (1952)

Kelly, Gene (I)

Boffo! Tinseltown's Bombs and Blockbusters (2006)

Bacon, Kevin

<u>Implementation Details</u>

The input text file consists of a number of lines consisting of '/' separated fields. The first field is a movie and all the remaining fields are the names of the actors and actresses what had a role in the movie. E.G. :

$20,000 Carat, The (1913)/Purdon, Richard/Moore, Tom (I)/Joyce, Alice

'Diamond S' Ranch, The (1912)/Maish, Frank/Mix, Tom/Mix, Olive

Batman Fights Dracula (1967)/Robles, Rolan/D'Salva, Ramon/Rivero, Dante/Nepomuceno, Nort/Heinsburg, Johannes Christof von/Abalos, Jing/Lorrian, Vivian

There are more than 285,000 movies and many more performers, the average vertex degree is relatively small so the logical choice is to use an adjacency list representation of the graph.

<u>Implementation notes</u>

The graph created using the movie data could be viewed in one of (at least) two ways. As either having a base class node from which two derived classes (actor/actress or movie) stem, or two separate graphs which are bound together by alternating links (i.e. all actor/actress nodes are only adjacent to movie nodes and all movie nodes are adjacent to only actor/actress nodes).

In addition to playing the Bacon number game, your project should be able to perform a variety of other operations on the graph once it's constructed. Since the data does not include weighted edges a simple BFS will be sufficient to calculate the all points shortest path from Kevin Bacon or any other specified node.

<u>Histogram</u>:

#	Freq
0	1
1	2249
2	217708
3	551279
4	100597
5	5286
6	466
7	17
Inf	56145

    Avg path length: 2.87515

Longest shortest paths

Be able to print out the end points for the performers with the longest-shortest paths (not including those who are not connected). Example: Tooran, Veisi is one of the performers with a Bacon number of 7 (which is one of the longest, connected, shortest paths to Kevin Bacon in the all06.txt data file).

```
Veisi, Tooran
Faisal, Khalid
Atashi
Pattovi, Ghodsi
Etc.
```

Other prolific performers:

Kevin Bacon isn't the only performer with high connectivity to other Hollywood performers. Set up your application so that any performer or movie can be used as the starting node for the game:

% Bacon_Number all06.txt "Raye, Martha"   % Bacon_Number all06.txt "Sound of Music, The (1965)"

0	1
1	1134
2	96140
3	572138
4	195866
5	11482
6	772
7	70
Inf	56145

Avg path length: 3.14018

0	1
1	28
2	19141
3	405518
4	415036
5	35597
6	2097
7	174
8	12
Inf	56144

Avg path length: 3.5402

The features of your program should include the following:

- Allow the user to produce shortest paths from any arbitrary performer or movie in the input file.
- It should accept any text file that is made up of records, one per line, with '/' delimited fields.
- It should have command line or menu options to produce a connectivity histogram (like above), a calculation of the average path based on any performer or movie, and the end-points for all the longest-shortest paths given an arbitrary starting point.
- Gracefully handle performer or movie names that are not present
  % Bacon_Number all06.txt "Weiss, John"
  Could not find performer/movie named [Weiss, John]

Usage details are up to you but your documentation must clearly describe the features and how a user would access them.

Notes

- A sample linux executable (Bacon_Number) will be posted on the website along with sample input files.
- You may use code from the CSC300 textbook or your instructor, but not other sources. Remember: always give credit when you use code written by other people.
- To receive full credit, your code must be readable, modular, well documented, and reasonably efficient, as well as correct. It must compile successfully under g++, the C++ compiler available in our Linux lab. If your program does not run correctly, indicate why. This will make it easier to give you partial credit
- When you are finished writing, testing, and debugging your program, submit your source code in a zip or tar archive using the Submit It! link on the MCS Department Website. Submit only source code, not executable files or test data. The submit program is accessed via the MCS Web page (http://www.mcs.sdsmt.edu), by selecting the list item on the left entitled "Submit it!". Usage is self-explanatory: enter your name, choose the instructor and click "Select Instructor", choose the appropriate course, browse to the filename you wish to submit, and click "Upload".
- Submit your program by the due date (Monday, December 7th) in order to receive credit for this assignment. Late programs will not be accepted for partial credit unless prior arrangements have been made with the instructor. If you have any problems with the submit program, report them to your instructor and submit your program by email instead.
- This is an individual project so no teams will be assigned.