

# 2015-16 Intersession Comp Prob Solving

## Lab E - Six Degrees of Kevin Bacon

### 1 Problem

You may have heard of the game or concept of “Six Degrees of Separation” or “Six Degrees of Kevin Bacon.” The idea in the latter is that any actor in any movie can be linked to Kevin Bacon through a series of actors that appeared in a movie together. For example, Ian McKellen (who played Gandalf in the “Lord of the Rings” trilogy) was in “The DaVinci Code” with Tom Hanks, who was in “Apollo 13” with Kevin Bacon. Supposedly all actors can be linked back to Kevin Bacon in no more than six links.

In this problem, program input is a list of movies and the set of actors that appeared together. The program needs to find a chain of no more than **three** links (rather than six, to keep the input files of a reasonable size) that connects two given names (*not necessarily Kevin Bacon*). Each line is in the format: “<Movie> <List of actors>”. Input files will look something like this example:

```
Apollo13 Kevin Bacon Tom Hanks Gary Sinise
HollowMan Elisabeth Shue Kevin Bacon Josh Brolin
AFewGoodMen Tom Cruise Demi Moore Jack Nicholson Kevin Bacon
OneCrazySummer John Cusack Demi Moore
DaVinciCode Tom Hanks Ian McKellen Audrey Tautou
```

For this assignment, the title is always be one “word”, and all names are exactly two “words” long. For the above data file and the actor name inputs John Cusack and Jack Nicholson, the output will have two links and look like:

Here is an example of running the program:

```
python3 sixdegrees.py
Enter data filename: test.txt
... # here the program prints its representation of the data.

Enter starting node name: John Cusack
Enter goal node name: Jack Nicholson

John Cusack
    was in OneCrazySummer with
Demi Moore
    was in AFewGoodMen with
Jack Nicholson
```

## 2 Submission and Grading

**Please make sure to submit all of the required components.** Zip all required files and submit the labE.zip file to the MyCourses dropbox. Your *individual* grade will be based on the following:

### 80%: Implementation

You will *individually* design and implement a program, `sixdegrees.py` that:

1. Prompts the user for the name of an input file.
2. Reads the file contents into a data structure.
3. Prints the contents of the data structure. *The output format is up to you, but all data from the input file must be accounted for in a readable manner.*
4. Prompts the user for two names to try connecting.
5. Performs the search and prints the shortest chain linking the two names.  
If the search finds no chain of length three or less, or no chain at all, the program prints “No chain exists”.

### 15%: Test Design

There are many possible inputs and search scenarios for this problem. In addition to your implementation, you will design and submit at least one additional test input file and a list of test cases to be run using this input. Samples will be provided via MyCourses.

1. `{test<N>.txt}` is a set of one or more files where `<N>` is a number starting with 1; these are files you used to test your program. You must submit at least one file that you used for testing.
2. `report.txt` is a plain text file that provides a list of at least 5 *unique* test cases that validate graph creation and the search. Each test case must have these elements:
  - (a) Test #
  - (b) Description of the test case (a short phrase is acceptable),
  - (c) Input file name, which must be a test file you submitted,
  - (d) Start actor name,
  - (e) Finish actor name, and
  - (f) Expected results or output.

### 5%: Design and Documentation

In addition to being graded on the working implementation and test cases, your program will be assessed for the following:

- Clean, readable source code that adheres to the CS department coding standards.<sup>1</sup>
- Appropriate use of variables, parameters and functions to minimize code reuse.
- Clear documentation of each program and its internal functions (i.e. docstring compliant comments regarding usage, design and implementation details).

---

1. See `PythonRecommendations.txt` in MyCourses or [online](#).