

Due April 1, 11.59pm

This project will use IMDB's movie/actor dataset to demonstrate computing the Bacon number between two actors, defined as the path length from a source actor to a destination actor, with the path going through a sequence of movies and actors). The project will be done in 2 parts, (1) reading and displaying the actor/movie graph, and (2) solving the Bacon number problem and displaying the path on the graph.

In part 1(this project), you will read in the given dataset and display the graph using an adjacency list representation. You will use the BRIDGES system to display and demonstrate some basic operations on the graphs.

Dataset:

We will use a small version of IMDB's actor/movie dataset for this project, as in project 1. This will be provided as a text file, containing a set of actor-movie pairs.

Tasks.

Here you will do a set of preliminary tasks that will be useful to complete the full project. Your application should have the following components ready and tested(all using BRIDGES).

1. **Load Input Dataset.** Read in the actor-movie dataset. Since the data comes in as actor-movie pairs, actors and movies can occur multiple times in the data, thus, we need to **ensure that the graph only contains one instance of an actor or movie.**

- Use the **Java Hashmap**(<https://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html>) to avoid duplicates. Use the actor and movie name as the key to insert or find if an actor or movie exists in the hash map. If the actor or movie does not already exist, then a new graph node (vertex) is created, else the existing node is used to create the link between the actor and movie. For example

```
GraphAdjList gr;  
String actorName;  
if (!gr.getVertices().containsKey(actorName)  
    "add a new vertex to graph");
```

Note that the BRIDGES graph implementation uses Java Hashmaps to index into the vertices list. Each vertex is of type $SLelement;E_i$, so a linked list represents the set of edge terminating vertices.

- Your graph should contain a unique set of actors and movies, once all of the data has been read into memory.
2. You will implement the following operation by accepting user input from the keyboard(you must become familiar with the `SLelement` class that is used to implement the adjacency lists in the Bridges graph implementation):
 - (a) The user will type in an actor name or movie and you will highlight the actor and all the movies linked to the actor (you will traverse the linked list and change the colors of the links and nodes connected to the actor/movie vertex)

- (b) Repeat the same if a movie is specified - highlight actors pointed to by the movie.
- (c) actor and movie names contain no spaces and can be read in as a single string

Evaluation:

By interactive demo; a schedule will be set up to test your program by the teaching assistant.

Submission Requirements.

Turn in your source code to Canvas by the deadline; ensure it is well documented.