

Due Dec. 6, 11.55pm

This project will use IMDB's movie/actor dataset to compute the Bacon Number of an actor, which refers to the shortest path through the movie/actor graph that connects two actors – the so called ‘Six Degrees of Separation’ problem. You will use BRIDGES to display the path that leads from the actor to Kevin Bacon(or another chosen actor).

BRIDGES Relevant Calls

The BRIDGES API is accessible from

<http://bridgesuncc.github.io/doc/api/0.99.993/>

Refer to the HelloWorld example that illustrates how to create vertices(nodes) and edges in BRIDGES, as well as modify their attributes:

http://bridgesuncc.github.io/main/HelloWorld-Tutorial_part1/

To access the children of a graph node (needed in the traversal algorithms), refer to the Vertex class, in particular the Vertex.next() method, which can be called repeatedly to return the children of that node(becomes null, for termination).

Dataset:

We will use the same dataset of IMDB's actor/movie dataset as in the previous project .

Tasks.

1. Build the full graph as in project 5a[No credit].
2. Implement the BFS traversal algorithm on graphs(see text for details). You will need to use a queue(can use the BRIDGES queue data type). The BFS traversal algorithm is sufficient to compute the Bacon number since the graph is not weighted(assume the weight of all edges of the graph is 1). Thus the shortest path between two nodes in the graph will be always determined by the BFS traversal (not true for a weighted graph). As before you will maintain a distance function at each node of the graph. When a child is reached, its distance value is then updated. Get a good understanding of the Dijkstra's shortest path algorithm, and modify the BFS algorithm to produce the Bacon number value. You can use a prev[v] array to keep track of the parent of the visited nodes(v) in the graph; this will aid in determining the Bacon number path.
3. **Bacon Number and Path:** Allow the user to input an actor name. Your program will compute the Bacon number(path distance to 'Kevin Bacon' node) of that actor and display the path (use a unique color to display the nodes and edges. The edges can all be the same color, while the nodes can be color coded(not a requirement) based on the distance from the 'Kevin Bacon' node. On your console(within Eclipse), also print the Bacon number and the path of actors/movies that lead to the Kevin Bacon node.

4. Perform a BFS traversal starting from the 'Kevin Bacon' node. Keep track of the longest path (and Bacon Number) to any actor in the graph. Display the longest path (similar to (2)).

Evaluation:

You will do an interactive demo of your implementation, similar to Project 5a.

Submission Requirements.

Turn in your source code to Moodle; ensure it is well documented.