The final outcome of the project is that we were able to execute a user interface based on command line that carried out all the required functionalities of our code, with proper user input.

Search engine:

Our most basic functionality which utilized the convenient design of the standard map. The final result showed that all 4 types of our search functions worked when user input was received.

Updatable dataset functionality:

We can add more datasets during the runtime of the program with our helper make graph functions.

BFS Traversal :

Breadth first search is the traversal that we have decided to use in our program.

We managed to successfully traverse a simple graph by using BFS. So from the image above, we can observe that we have test cases for our functions that we use for our program implementation. For example, We have a test case for BFS_traversal. The way we check that the traversal was correct. The BFS_traversal is used for our traversal of choice for our force directed graph drawing.

Dijkstra's Single Source Shortest Path Algorithm:

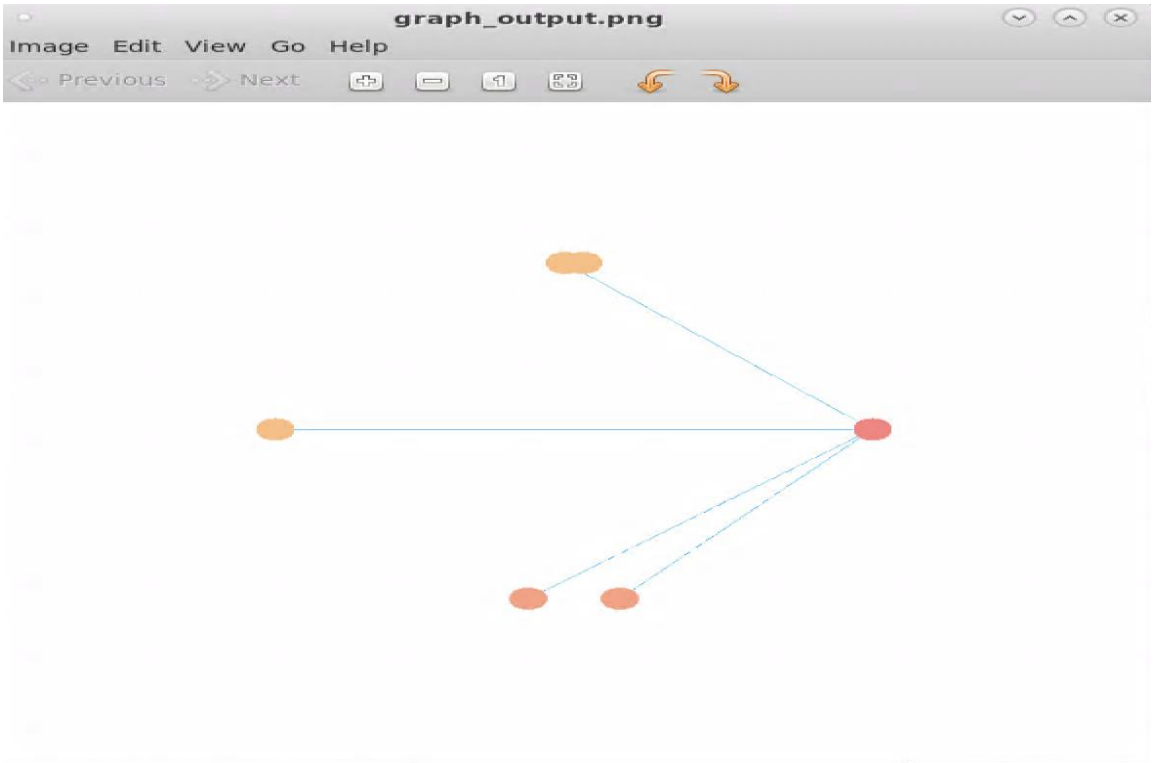
The dijkstra's algorithm is used in our program when the user wants to input two movies and requires our program to recommend movies that is related to those two inputs. The way the dijkstra's is used is to find a shortest path from the first argument (starting vertex) to the second argument (destination vertex). Since the vertices is connected by an edge if they are related (whether through genres or common actors), the vertices in the shortest path is going to be vertices that have the most common traits between the two argument vertices. The dijkstra function is also tested in our test.cpp by checking it whether the output to be same as the one that is hand calculated.d

Graphical Representation of the Graph Output:

One of the algorithms that we have implemented is the force-directed graph drawing algorithm, which we used to produce a graphical output of the graph that we traverse through. As seen below, we have tested the function that prints the PNG representation of the graph with simple graph with 2 different genres. As we can see, it produces line for wherever the edge is supposed to be and circles of different colors to represent vertices of different genres.

Recommendation generation:

We used dijkstras to properly find the shortest path on our graphs whose edge weights were defined by common genres and actors. Using this as the basis for our recommendations, we found the movies most similar to the given user input and hence considered them the best recommendations for the user. Our algorithm worked and was able to give out movie titles as recommendations.




1200 × 1200 pixels 3.2 kB 49% 1 / 1

```
*                Members include: ARSAM, HADI, RIDHWAN, MUSTAFA                *
*     This tool has multiple uses, which we will describe and list out the     *
*     prompts for activateing them as well in the following information:       *
*     USE 1:                                                                   *
*          DESCRIPTION: search engine, options for specifying search           *
*          criterias described after prompt is given                          *
*          PROMPT: type 1.                                                    *
*     USE 2:                                                                   *
*          DESCRIPTION: feed in data in the form of a csv file and have       *
*          the database expanded. More information on the format of the       *
*          file is given when prompt is typed.                                *
*          PROMPT: type 2.                                                    *
*     USE 3:                                                                   *
*          DESCRIPTION: ask for a recommendation on which movies to           *
*          watch and our algorithm will try to find the movies that match     *
*          your taste.                                                        *
*          To get the best result, have some search history in the            *
*          duration of the execution of this program                          *
*          PROMPT: type 3.                                                    *
*     USE 4:                                                                   *
*          DESCRIPTION: ask for a visual output of the movies stored in       *
*          the databse of the program and we will output a PNG which          *
*          contains a scatter of points representing movies clustered by      *
*          genre, each genre being a different color.                         *
*          PROMPT: type 4.                                                    *
*                                                                             *
*     To quit the program, type 5.                                            *
*                       TO VIEW THE LIST OF USES AGAIN, TYPE 6                *
*******************************************************************************
Waiting for prompt: 2
Format required: title, genres, actors, ratings
movie.csv
Waiting for prompt: 3
Enter 2 of your movies on which you wish to base the recommendations:
movie 1:
Sing
movie 2:
Split
Finding movie 1...
movie found!
Finding movie 2...
movie found!
```

```
/* Testing BFS_traversal() Function with 3 vertices graph */
============================================================

How the movies are ordered in the linked list.
Movie 1
Movie 2
Movie 3

Since makeSimpleGraphWithEdge() does not have an edge between Movie 1 and Movie 2,
the traversal should be Movie 1 -> Movie 3 -> Movie 2
Movie 1
Movie 3
Movie 2

/* Testing dijkstrasl() function with 4 vertices graph */
=======================================================

The start point is Movie 1 and the destination is Movie 4

Since the ouput is inverted, the output should be:
Movie4->Movie2->Movie3->Movie1


/* Testing dijkstrasl() function with 3 disconnect vertices graph */
==================================================================

The start point is Movie 1 and the destination is Movie 4

Since the vertices is disconnected, the output should be:
[The two vertices is disconnected.]

The two vertices is disconnected.

/* Testing I/O interface */
=========================
Guardians of the Galaxy
Action
Adventure
Sci-Fi
Chris Pratt
Vin Diesel
Bradley Cooper
```