## Introduction

Recommendation systems in today’s world of abundant information is a very powerful tool that can be helpful to consumers of all sorts. Whether you are trying to find the next movie, game or a television series to enjoy, want to know where the cheapest restaurant near you that is opened at 3am or if you wish to figure out which keyboard to buy that fits your needs, a recommendation system can make it all quite hassle free without you needing to research because the data provided to the algorithm will try to do all that for you.

Amazon, Netflix and Steam are some of the biggest organizations that are looking to make their recommender system even a tiny percentage better as even a small increase can bring in a lot of revenue. Therefore, we thought this project would be a great way to mimic those much bigger projects and to see for ourselves how they work.

## Data structure implemented

* **Graph**: We implemented undirected, weighted graph data structure represented in the form of adjacency matrix. This data structure is the backbone of this project as the movies that are recommended have been implemented as nodes.
* **Arrays**: We created an array consisting of objects of movies that contain further information about the movies such as their category and the director.

### Inner workings

The project consists of three distinct parts :

* Movie list: for this program the list of movies has already been hard coded into the program, this list is an array called movieList.
* Movies watched by the user: the user tells the program what movies they have watched and out of those which ones he liked. The program stores these movies in an array called moviesWatched. The array is also containing movie objects that are directly copied from the movieList array.
* Similarity point: the user is now recommended the movie they should watch depending upon which two nodes had the highest ‘length’. It then recommends the movie with the highest similarity point.

### Participation

Although we tried, there was no determinate way to figure out who did what part as the pseudocode as well as the code was written within a single day. We spent the whole day together and discussed back and forth about how it should be implemented and then took turns coding on a single laptop. We sincerely do mean it when we say that we both participated to the best of our abilities and would like to say we both did 50% each.

### References

### https://www.youtube.com/watch?v=ZspR5PZemcs