Oluwatobi Oyinloye

DS 210 Project - Final Report

Harry Potter Character Network

Project Topic

I decided to implement the Six Degrees of Separation theory for my project, using a dataset of every character in the Harry Potter universe and all other characters they've directly interacted with throughout the series. I thought this would be an interesting topic to work on, especially for such a vast character network like Harry Potter. The Six Degrees of Separation theory says that any two people on Earth are six or fewer links apart. With a world like Harry Potter, I think this to be very true and possibly on average, less than six links apart. My program aims to randomly select a character, analyze all its interactions and deliver an output which shows its level of degree of separation with every other character.

Dataset:

The choice of Harry Potter characters for this project was due to the large number + network of them. Unfortunately I couldn't find a dataset that met my needs but, I found this JSON : https://github.com/dpmartin42/Networks/blob/master/Harry%20Potter/data/HP_network_clean.json

I manually made it into a table (with vertices and nodes) for every character and their interactions, then converted it into a CSV which I used for the project.

Code:

csv_to_txt

This module starts the data processing stage by reading data from a csv file. It also cleans the data by removing duplicates then writes the cleaned data into a txt file. In the main function, this module will output the new txt file and "CSV to TXT successful!" sentence.

read_csv

This module opens the csv and iterates through each line, turning it into nodes. It also filters out lines that don't contain at least two elements. Then, it returns a vector of (String, String) tuples representing edges extracted from the CSV file.

remove_duplicates and write_txt

This module uses a hash map (unique_edges) to store tuples (node1, node2) taken from edges, which it then inserts each edge tuple into the HashMap, overwriting duplicates. Its results are returned as a vector of edges.

load_data

This module is responsible for reading through the data and uses node_map to track node indices for each character. It also adds edges between each node for characters depending on their relationships. It's output in the main function is text that says "Graph loaded successfully!"

random_sample

This module takes a slice of items and a sample_size as input then uses rand::thread_rng() to initialize a random number generator. It's purpose is to randomly select a specified number of characters.

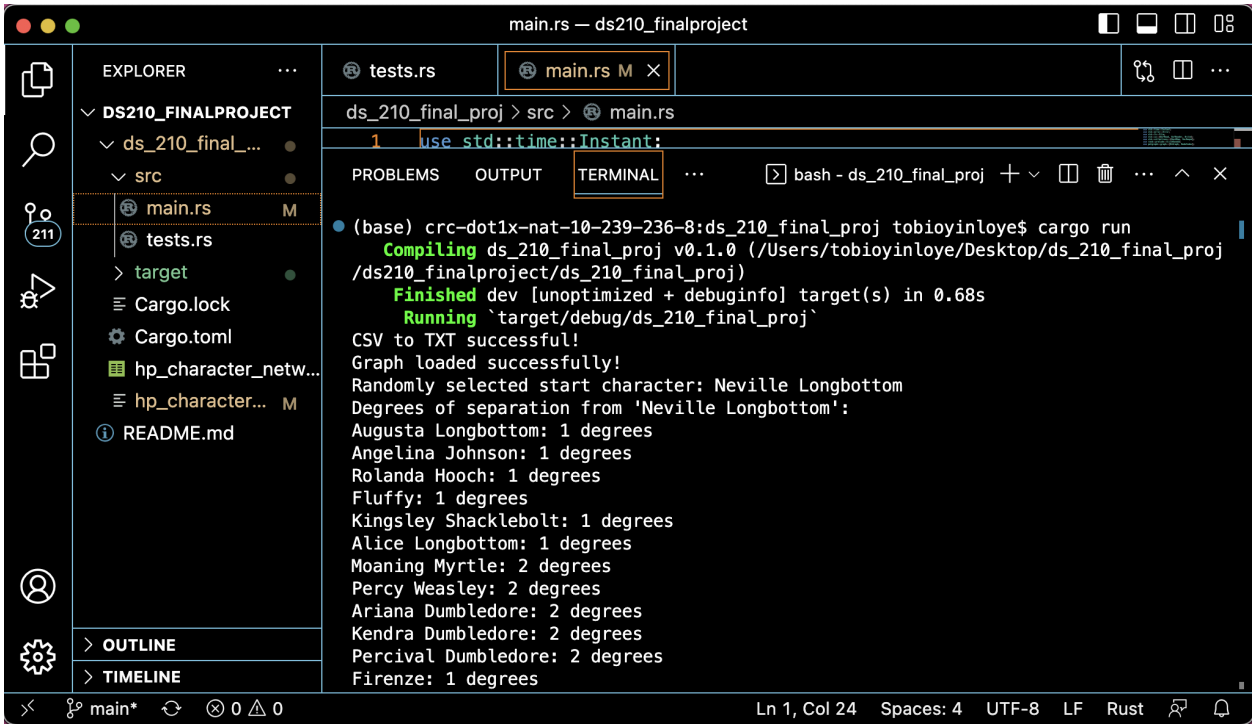
six_degrees_to_all

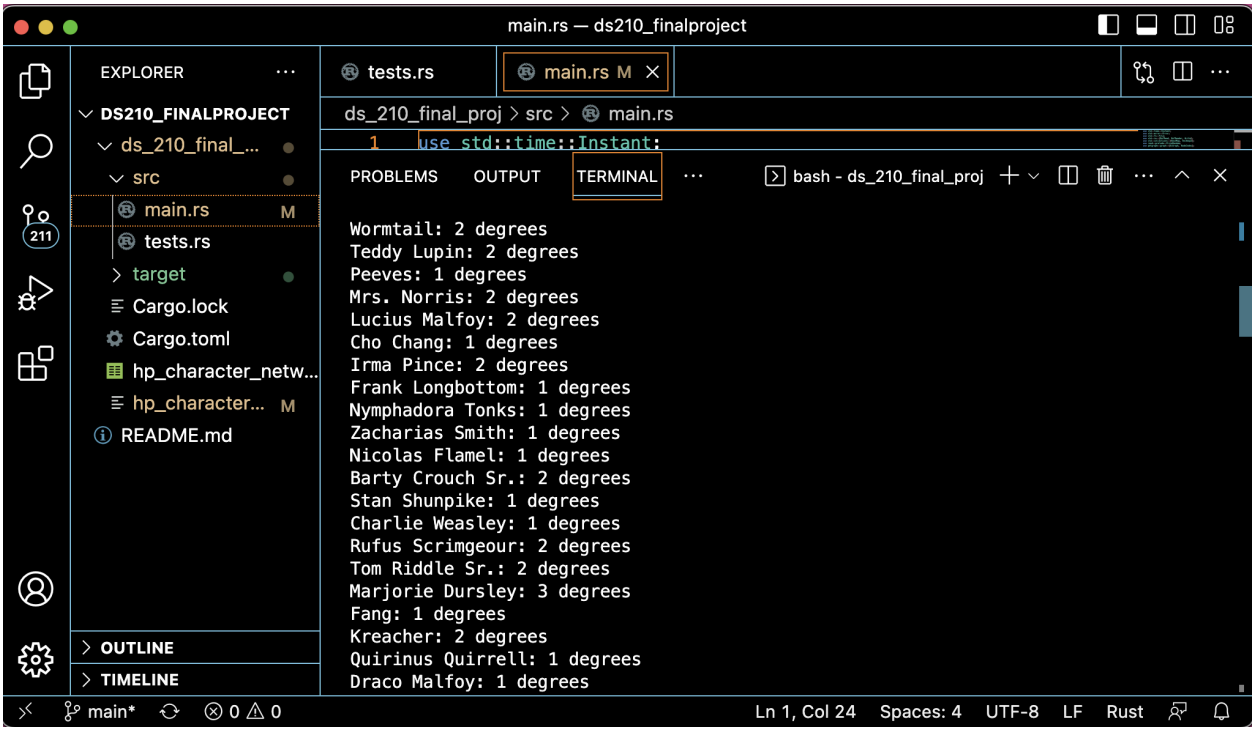
This is the core of the Six Degrees of Separation theory and uses the breadth-first search (BFS) algorithm to calculate the degree of separation from a starting character to all the other characters in the graph.

main

The main function is a compilation of the modules above, leaving the space for me to manually input the name/file path of my csv file as well as what I want the txt.file to be named as. It outputs the name of the randomly selected character and it's relationships with the rest. It also outputs the time it took for the algorithm to run (in microseconds). In general, my results indicate that all characters in the Harry Potter network are indeed linked in some way. One way I would improve on this algorithm is to write a module that calculates the average degree of separation for all the characters as well as which character has the highest - it would be interesting for statistical purposes. Here's what my output looks like:

Oluwatobi Oyinloye





Oluwatobi Oyinloye