Louie Belile

I had decided to use the Marvel database of hero co-occurrences within Marvel comics and movies in my project. I wanted to analyze the importance of each character within the Marvel universe by creating a network graph. My dataset contained 574,468 lines, where each line is compiled with two heros separated by commas and quotations, example line→“Hero1”, “Hero2”. I created an undirected graph of the co-occurrences which breaks down to 574,467 edges and 6421 vertices. The vertices represent the heroes/characters, and the edges represent the co-occurrences between them. An edge between two heroes means that they have appeared together in at least one comic book.

In the graph module, the graph is represented using an adjacency list data structure. The Graph struct contains the number of vertices n, a vector of adjacency lists outedges representing the edges leaving each vertex, a vector of vertex labels containing the label of each vertex, and a hashmap called vertex_indices mapping vertex labels to their indices in vertex_labels. Within the the graph module is the he compute_and_print_centralities method. It computes and prints the centrality of each vertex in the given graph. It initializes a vector centrality to store the centrality of each vertex, and then for each vertex s, it performs a breadth-first search to compute the distance and number of shortest paths from s to every other vertex. It then computes the centrality of s by summing the number of shortest paths from s to each other vertex that is one hop away from s and dividing by the total number of shortest paths from s to any other vertex. It stores the resulting centrality in centrality and prints it. Finally, it sorts the vertices by centrality and prints the top three vertices with highest centrality.

There was also a test module to test that the read file module works as intended as reading the data correctly is the only way to ensure that the other methods will work. When this code is ran with the marvel dataset and “cargo run —release” is used as the command, it takes about 4.5 minutes to run.

My Discoveries: Captain America had the highest centrality score meaning he was the most frequent occurrence in the shortest paths which means it is the most important character in the universe social network graph. Spiderman came in second and Iron-Man came in third.

Example output: