Project Report

Tye Robison

For this project I utilized a Marvel Universe dataset containing every marvel hero and the comics they have appeared in. My intent was to determine which characters are the most prominent in all of the comics and use the 6 degrees of separation algorithm to see which characters can be reached with 6 degrees, the average amount of connections between each character, and more. I really love the Marvel Universe and was interested in determining which heroes are the most important and most prominent.

In order to do this, I began by importing the dataset, reading the file, and creating a graph of the heroes and the associated comics. I used a module, entitled read.rs, to read the file and create an empty hashmap and then proceed to fill the hashmap with the heroes and their corresponding comics.

After this I worked on implementing MarvelGraph and the functions within it in another module entitled graph.rs. To begin, I created the function from_comics_data to create the MarvelGraph which looks at the entire data set and established the connections between each hero and the similar comics between them; this part was particularly difficult because originally it was counting each appearance of a hero in the data set as 1 node instead of connecting the heroes name to a specific comic (aka it did not avoid repeats), but eventually I figured this out and added edges between all heroes in the same comic and which checks itself to ensure the edge doesn't already exist.

Then I introduced the average_connections functions in my implementation which is self-explanatory in that it tells us the average connections between all characters. The next function computes centrality and which then calculates the top 5 heroes with the highest centrality based on their connections with other heroes. After this, I created a function which finds the hero with the most connections based on the most comic appearances. Finally, my last

function in this module calculates the amount of reachable heroes within 6 degrees of separation. This function uses an algorithm called Breadth-First-Search in order to iterate over all the neighbors for each hero in the dataset and then checks the degrees of separation and collects these lists in a hashmap where each key is a hero's NodeIndex and each character that can be reached within 6 degrees is recorded as a set of NodeIndex values.

Regarding testing, I have three tests written into this project in my main.rs file. Each test works to confirm a different aspect of the code. The first test serves to check if the graph is created properly using sample data which ensures the edges and nodes properly link together using my pre-established functions. Then, a second test is used to ensure the most_connected_hero function works correctly by using more sample data to test the function once again. Lastly, I wanted to test arguably the most important aspect of the code, the degrees_of_separation function. This uses sample data and my compute_reachable_heros functions to check that the implementation of the 6 degrees of separation is working correctly.

As far as the output is concerned, I was able to gain a lot of knowledge from the dataset. As explained earlier, the first output we see is the top five heroes who have been in the highest percentages of comics. This is interesting considering the hero with the most connections to other characters is Spider-Man, yet he is not on the list of the top 5. I believe this is the case because comics containing Spider-Man usually include many hero families due to his connection with Tony Stark and the avengers. Also, in many comics Spider-Man appears as a supporting character to the primary hero due to his popularity among fans. Also, I took the time to calculate the percentage of heroes connected within 6 degrees which came out to be 32.77% of characters; this was done using the 6 degrees of separation algorithm on all characters and their neighbors. Finally, the last calculation is used to determine the average number of connections each superhero has, which evened out to 26.65, so almost 27 connections per character.

Output/Tests

```
(base) tyerobison@crc-dot1x-nat-10-239-160-39 project1 % cargo test
   Compiling project1 v0.1.0 (/Users/tyerobison/Documents/project1)
    Finished test [unoptimized + debuginfo] target(s) in 2.22s
     Running unittests src/main.rs (target/debug/deps/project1-f7637e7afee4f7a0)

running 3 tests
test tests::test_graph_creation ... ok
test tests::test_most_connected_hero ... ok
test tests::test_degrees_of_separation_with_precomputed_data ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

(base) tyerobison@crc-dot1x-nat-10-239-160-39 project1 % cargo run
   Compiling project1 v0.1.0 (/Users/tyerobison/Documents/project1)
    Finished dev [unoptimized + debuginfo] target(s) in 0.27s
     Running `target/debug/project1`
Top 5 heroes in terms of the percentage of comics they have appeared in:
CAPTAIN AMERICA: 25.94%
BEAST/HENRY &HANK& P: 19.35%
ANGEL/WARREN KENNETH: 17.24%
ANT-MAN/DR. HENRY J.: 16.99%
IRON MAN/TONY STARK: 14.37%
The hero who has the most connections is: SPIDER-MAN/PETER PARKER
Percentage of heroes connected within 6 degrees: 32.77%
Average connections per character: 26.65
```

Sources

- The Marvel Universe Social Network
- https://docs.rs/petgraph/latest/petgraph/
- https://doc.rust-lang.org/rust-by-example/mod/visibility.html
- https://doc.rust-lang.org/reference/items/use-declarations.html
- https://web.mit.edu/rust-lang_v1.25/arch/amd64_ubuntu1404/share/doc/rust/html/book/second-edition/ch07-02-controlling-visibility-with-pub.html
- https://marketsplash.com/tutorials/rust/rust-error-types/
- https://codereview.stackexchange.com/questions/283176/rust-implementation-of-multiple-type-of-graphs-with-shared-code
- https://doc.rust-lang.org/std/collections/struct.HashMap.html
- https://docs.python.org/3/library/itertools.html
- https://docs.rs/petgraph/latest/petgraph/visit/trait.VisitMap.html