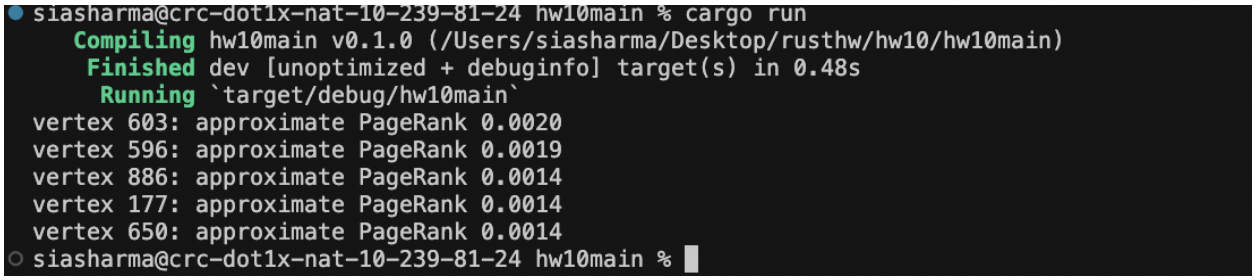
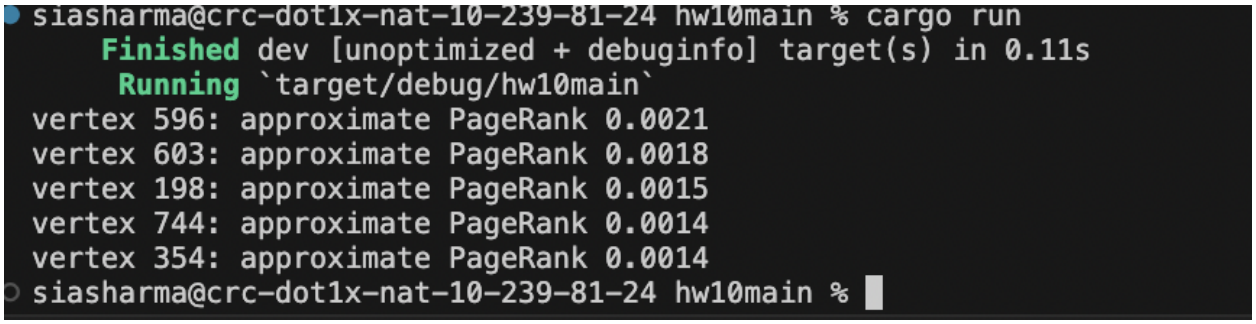
Sia Sharma
Collaborators: Om Italiya, Nikita Salkar

Code Output:



When it is run again, this is the output:



Analysis on my program:

- My program calculates PageRank values for vertices in a graph. The program begins by reading graph data from a file, where the first line specifies the number of vertices, and subsequent lines represent edges between vertices. Using a random walk algorithm, I simulated transitions between vertices to approximate PageRank values. In the random_moves function, I conducted random walks on the graph, with each walk starting from a vertex and moving to neighboring vertices based on the probabilities given in the HW set. During each walk, I counted the number of times each vertex is visited, storing these counts in a vector. After simulating a large number of random walks, I calculated the PageRank values for each vertex based on the visit counts.
- To identify the top vertices with the highest PageRank values, I sorted the PageRank values in descending order and associated each value with its corresponding vertex. I then printed the top vertices along with their respective PageRank values.
- I also implemented error handling when reading the graph file, which printed error messages if necessary, for which I used help from https://chat.openai.com/. My test function makes sure the graph reading process is correct.
- Throughout the code, I used the rand crate for generating random numbers and making random choices during the random walk simulation.