**Page Rank Documentation**

Data Structure: For this project I utilized an adjacency list class. This class was functionally a wrapper class that encompassed a map, where the key was a string representing the from page, and the value was a list containing all the to pages that the from page linked to. A map was used as the key, value construct was important to associate the from page to the pages it linked to. A list was used to store the to pages as it was the lightest weight container that allowed storage, iteration, and stored a size variable.

Time complexity, methods:

The adjacency list implements two public methods:

Insert: inserts the from page and to page of an edge into the underlying map. Runs in constant time as there is no iteration.

Size: retrieves the size variable from the underlying map, runs in constant time.

A begin and end method are also implemented that return the beginning and end of the underlying map. An iteration through the keys in the adjacency list runs in O(|V|), though the pagerank algorithm also iterates through the values in the map, which makes this version of an iteration run in O(|E|)

The PageRank class implements three methods:

ParseInput: gets the input from the user and stores the input edges into an adjacency list. Runs in O(|E|).

IteratePageRank: Runs the actual algorithm. Iterates through all edges in the adjacency list p times, where p is the power iteration input by the user. Results in O(p \* |E|).

PrintResults: Outputs the results of the page rank algorithm. Runs in O(|V|).

The time complexity of the main method (IteratePageRank) runs in O(p \* |E|), where p is the power iteration number, as discussed above.

Lessons learned: RTFM. During project one, I spent a significant chunk of time attempting to figure out how to programmatically test the project with IO. Never figured it out and missed two IO related test cases because of it. Spent more time during this project trying to figure it out and got nowhere. Randomly clicked in the readme file included in the Catch code for project one and discovered that the answer that I need was readily available to me the entire time. So, RTFM, and make sure to go through all the given materials at the beginning.