

Project PageRank

Patrick Leimer 88717127

Describe the data structure you used to implement the graph and why. [2.5 points]

For the graph, I implement the adjacency list using a map, with each key being a website and having a vector of strings that contain the edges/connections from the website to the others. I used because it seemed like the most efficient and viable way to do so, reason is that with the adjacency matrix there might be a lot of wasted space if there are not a lot of edges. For that reason, the list turns out to be the best solution. I also used a set to gather up all the unique websites within the graph.

What is the computational complexity of each method in your implementation in the worst case in terms of Big O notation? [5 points]

Using a set and a map, both of these have $O(\log V)$ in worst case. For the constructor, inserting into map or set in this case that is sorted and has n keys, then is $O(V * \log V)$, and then inserting outbound websites to it should be $O(E * \log V)$ which makes the worst case complexity $O((V + E) * \log V)$. Now for the PageRank method, we have n power iterations in which worst case for the nested for loops is going to be $O(V^2)$.

What is the computational complexity of your main method in your implementation in the worst case in terms of Big O notation? [5 points]

Main method has a for loop that iterates through the lines (n lines) and then the map the we insert the url's into has a $O(\log V)$, so worst case should be $O(n * \log V)$.

What did you learn from this assignment, and what would you do differently if you had to start over? [2 points]

I honestly learned so much, data structures, hash maps, sets, and how we can approach these algorithms in so many different ways. Adjacency list or matrix, benefits and cons of using these data structures. Overall a great learning curve. I wouldn't do anything differently, just was very sick over the weekend and didn't submit it in time.