Nicholas Miller

Data Structures and Algorithms

Page Rank Write Up

**Table Data Structure:**

The data structure I decided to implement my table of URLs with was C++’s STL for Map. I choose this structure because it allowed me easily map URLs to a specific ID and access them. If I were to restart the project I probably wouldn’t choose a different data structure to implement the table of URLs because was C++’s STL for Map is such a great option.

**Graph Implementation:**

For this project, I decided to implement my graph using an adjacency matrix. I choose this implementation because I knew I was going to be doing a lot of accessing within the matrix when matrix multiplying. THIs implementation gave me a constant complexity which allow for efficient matrix multiplying. If I were to restart the project I may try to implement my graph using an adjacency list instead.

**Computational Complexity:**

Insert Edge and Get Weight:

The complexity of insert edge and get weight is O(1).

Graph Constructor and Destructor:

The complexity of Constructor and Destructor is O(|V|).

Matrix Multiply and Print Graph:

The complexity of matrix multiply and print graph is O(|V|^2).

Obtaining and Storing Information:

The complexity of obtaining and storing information is O(n), where n is the number of links plus the number of URLs.

**Hardest Part:**

The hardest portion of the project was finding the values I was going to insert into my graph and where to insert them. It took me awhile to understand the project descriptions, but I eventually understood what I needed to do. After that the rest of the project was relatively easy.

**What I Learned:**

I learned two things from this project. The first thing was an actual use for a map. I always understood the concepts of a map but never saw a great use for it in programming. The second thing I learned was how to implement a graph. I hadn’t implemented a graph before and it was a fun experience doing so. I learned a lot about the moving parts of a graph which are going to help me a lot in the future.