

1. Data structure of choice

I implemented the graph through using one map and two unordered maps.

- `Unordered_map<string, unsigned int> outgoingLinks` – This map is responsible for keeping track of the outdegree which is the number of outgoing edges for each vertex. Unordered map provides a $O(1)$ access, which retrieves and updates the data efficiently.
- `Unordered_map<string, vector<string>> incomingLinks` - This map stores the incoming links for each vertex, the vertex maps to a vector of strings which show the vertices that are linked to it. This allows for quick access for all the incoming links.
- `Map<string, float> PageRank`: This map stores the PageRank value of the vertex. Since it's sorted on their keys for order purposes, I chose a map which keeps the order. This has $O(\log n)$ access and update time.

2. Computational Complexity of each method

1. InsertEdge Method:

- Adds edge which updates outgoingLinks and incomingLinks
- Each update in an unordered is $O(1)$ on average.
- The overall complexity is $O(1)$.

2. InitializePageRank Method:

- The Method goes through each vertex to set the page rank value initially.
- The complexity is $O(V)$, where V is the number of vertices.

3. PageRank Method:

- The method runs power iterations times, which goes through all the vertices for all the iterations
- For each iteration, each vertex with incoming links it goes through each link to calculate the sum.
- Complexity for each iteration: $O(V+E)$
- Overall complexity is $O(\text{power_iterations} * (V+E))$

4. printPageRanks Method:

- This method goes through the PageRank map and prints out each value
- The complexity is $O(V)$

3. Computational Complexity of Main Method

- Reading Inputs: Reading lines and power iteration is $O(1)$ and reading each edge both from and to is $O(E)$.
- Inserting Edges: For each edge, you call insertEdge, which is $O(1)$. This inserts all edges which is $O(E)$.
- Calling Page Rank: Calling the PageRank methods is $O(\text{power_iterations} * (V+E))$. So the worst case complexity of main method would have to be $O(E + \text{power_iterations} * (V+E))$.

4. Overall Reflection

This assignment taught me about how to structure the PageRank algorithm and the mechanics involve. How to represent directed graphs in code. Working with unordered maps and maps in order helped optimize the operations. If I had to start over, I would go into sparse matrix representation so I would be able to store the graph for large data, this would let me be able to have more efficient memory usage. This would allow for better memory usage.