Search Engine Report

Ali Khattab

900191895

1. **Pseudo code for ranking:**

* Calculate ctr() { click\_through/impression \* 100%}
* Get\_pagerank()

{

Initializes pr for every website to be 1/n

initialize double array[n] to store previous iterations

Loop 5 times (number of iterations to be done) **5\*O(…)**

{

Store pr of every website into previous iterations array **O(n)**

Loop over all the websites **O(n)**

{

Double sum;

Loop over adjacency matrix to find the websites that point to this website **O(m)**

Sum = where n stands for number of websites pointing to website **O(2)**

}

Assigns pr of website to be the calculated sum  
}

Update previous iteration to be the same as the final iteration **O(n)**

quick sort previous iteration from highest to lowest **O(nlogn)**

Calculates final pr for every website to be ranked from n to 1 according to order of sorted arr **O(n)**

}

* Calculate score () {equation given for calculating score using ctr and pr}

1. **Time complexity:**

* Ctr() = **O(2)**
* Page\_rank() = 5(n + n(2m) + 1) + n + n^2 + n = 5(n+2nm + 1) +2n +nlogn = 5n + 10nm + 5 + 2n + nlogn = 10nm + 7n + nlogn + 5 = **O(nlogn)**
* Score() = **O(13)**

**Total time complexity = O(nlogn)**

1. **Data structures:**

* Adjacency list was made with vector of vectors of type website nodes, but this was only for dealing with the page ranks to deal with paths and incoming/outgoing edges
* double[n] was used to store previous iterations
* vector of websites of size n was used to store the given page rank in
* In terms of the actual search engine, the vector of websites was the only data structure used since we had no need for the graph after getting the website scores