Project Report

Michael Henein – 900212000

Time and space complexity analysis:

The search runs in O(n \* m) in the worst case scenario given that the user enters AND with n being the number of sites that have the first key word and m being the number of sites that contains the second key word.

The page rank algorithm runs in O(n^2) since it is three nested loops with the outer most loop being constant at 30. The second nested loop passes through all the vertices of the graph and the third nested loop passing over all the vertices of the graph.

Data structures used:

Built in hashing algorithms using sets and maps.

Storing the graph in an adjacency matrix.

Vectors.

Pseudo code:

Ranking algorithm:

for (all verticies of the graph) {

initialize rank with 1 / 30

}

double max, min;

for (int i = 0 -> 30) {

max = 0; min = 1.0;

for (all verticies of the graph) {

double sum = 0;

for (all other verticies of the graph) {

if (the second vertices has an edge directed at the first) {

the rank of the second node divided by the number of verticies that it points to are added to the sum

}

}

The rank of the first node is updated to the sum

if (sum < min) { min = sum; }

if (sum > max) { max = sum; }

}

}

for (all verticies of the graph) {

their normalized ranks are calculated using ((rank - min) / (max - min))

}