Mohamed Wael Shaalan

900201539

CSCE2203

Project Report

Pseudocode

Calculating Rank: (getSortedWebsitevector())

vector<vector<double>> **pageranktable**; (websitenumber*websitenumber)

vector <double> **currentrow**;

set initial row of **pageranktable** as 1/(number of websites);

for(looping over pagerank table ignoring first row)

{

update **incomingwebsites** by using second loop index;

update **currentrank** by looping over **incomingwebsites** and using the page rank summation formula with each loop adding one iteration of the page rank summation to current rank;

pushback **currentrank** to **current** row;

}

pushback **currentrow** to **pagerank** table;

using **setpagerank** to **websitevector** by accessing final row of **pageranktable**;

Indexing Websites: (sortByscore())

vector<website> websitevector; (websites are read and then inserted into a vector of object type website based on order in files;

Sort by looping over websitevector using bubblesort descendingly.

Complexity Analysis

n=*incomingwebsite.size();*

m=*websitevector.size();*

Calculating Rank:

Vector Accesses:

$$\sum_{1}^{m}\sum_{j=0}^{m}\sum_{k=0}^{n} 3 = \sum_{1}^{m}\sum_{j=0}^{m} 3(n+1) = \sum_{1}^{m} 3(n+1)(m+1) = 3(n+1)(m+1)(m)$$

Complexity =O(nm^2)

Arithmetic Operations:

$$\sum_{1}^{m}\sum_{j=0}^{m}\sum_{k=0}^{n} 2 = \sum_{1}^{m}\sum_{j=0}^{m} 2(n+1) = \sum_{1}^{m} 2(n+1)(m+1) = 2(n+1)(m+1)(m)$$

Complexity =O(nm^2)

Indexing Websites:

$$\sum_{0}^{m} 1 = 1(m+1)$$

Complexity =O(m)

Data structures used:

1) Vectors
2) Graph
3) Adjacency Matrix