|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Finolex Academy of Management and Technology, Ratnagiri** | | | | | | | | | |
| **Department of Information Technology** | | | | | | | | | |
| Subject name: Big Data Lab | | | | | | | | Subject Code: ITC801 | | | |
| Class | | BE IT | | Semester – VIII (CBSGS) | | | | Academic year: 2019-20 | | | |
| Name of Student | | **Kazi Jawwad A Rahim** | | | | | **QUIZ Score :** | | | | |
| Roll No | | **28** | | | Assignment/Experiment No. | | | | | 07 | |
| Title:  **Implementation of Page rank Algorithm** | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **1. Course objectives applicable:**  **COB4**.Study Page Rank in Link Analysis and concepts of Handling larger datasets | | | | | | | | | | | |
| **2. Course outcomes applicable:**  **CO4**-Implement use of combiners to consolidate results and ability to handle larger datasets | | | | | | | | | | | |
| **3. Learning Objectives:**   1. To understand concept of Page rank 2. To understand teleportation factor 3. To program Page rank computation in C/C++ 4. To prove that teleportation helps to counteract problems caused by dead ends and spider traps | | | | | | | | | | | |
| **4. Practical applications of the assignment/experiment: Page rank is used by Google Search engine for indexing of webpages and giving results for search queries** | | | | | | | | | | | |
| **5. Prerequisites**:   1. Understanding of Internet Technologies | | | | | | | | | | | |
| **6. Hardware Requirements**:   1. PC with 4GB RAM, 500GB HDD,   **7. Software Requirements:**  1. Access to C/C++ compiler  2. Internet access if online compiler is used | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)**   1. What is a Page rank? 2. What is Teleportation? 3. What is a dead end? 4. What is Random Surfer? | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **9. Experiment/Assignment Evaluation:** | | | | | | | | | | | |
| **Sr. No.** | **Parameters** | | | | | | | | **Marks obtained** | | **Out of** |
| **1** | Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used - | | | | | | | |  | | 6 |
| **2** | Neatness/presentation | | | | | | | |  | | 2 |
| **3** | Punctuality | | | | | | | |  | | 2 |
| **Date of performance (DOP)** | | |  | | | **Total marks obtained** | | |  | | **10** |
| **Date of checking (DOC)** | | |  | | | **Signature of teacher** | | | | | |

**12. Installation Steps / Performance Steps –**

**PageRank Program with Teleportation:**

#include<iostream>

using namespace std;

int main ()

{

int i, j, k,n;

float A[7][7]={0,0,0,0.33,0,0,0,1,0,0,0,0.5,0,0,0,1,0,0,0.5,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.33,0,0,0,0,0,0,0.33,0,0,1};

float B[7][1]={0.143,0.143,0.143,0.143,0.143,0.143,0.143};

float T[7][1],C1[7][1]={0};

for (i = 0; i < 7; i++)

{

for (j = 0; j < 1; j++)

{

C1[i][j] = 0;

for (k = 0; k <7; k++)

{

C1[i][j] += ((0.8\*A[i][k])+0.03) \* B[k][j];

}

}

}

cout << "\nProduct of matrices ";

for (i = 0; i < 7; i++)

{

for (j = 0; j < 1; j++)

cout << C1[i][j] << ", ";

}

for(n=0;n<60;n++)

{

for (i = 0; i < 7; i++)

{

for (j = 0; j < 1; j++)

{

T[i][j] = 0;

for (k = 0; k <7; k++)

{

// T[i][j] +=(A[i][k] \* C1[k][j]);

T[i][j] += ((0.8\*A[i][k])+0.03) \* C1[k][j];

}

}

}

cout << "\nProduct of matrices ";

for (i = 0; i < 7; i++)

{

for (j = 0; j < 1; j++)

{

cout << T[i][j] << ", ";

C1[i][j]=T[i][j];

}

}

}

return 0;

}

**PageRank without Teleportation:**

#include<iostream>

using namespace std;

int main ()

{

int i, j, k,n;

float A[7][7]={0,0,0,0.33,0,0,0,1,0,0,0,0.5,0,0,0,1,0,0,0.5,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.33,0,0,0,0,0,0,0.33,0,0,1};

float B[7][1]={0.143,0.143,0.143,0.143,0.143,0.143,0.143};

float T[7][1],C1[7][1]={0};

for (i = 0; i < 7; i++)

{

for (j = 0; j < 1; j++)

{

C1[i][j] = 0;

for (k = 0; k <7; k++)

{

C1[i][j] += A[i][k] \* B[k][j];

}

}

}

cout << "\nProduct of matrices ";

for (i = 0; i < 7; i++)

{

for (j = 0; j < 1; j++)

cout << C1[i][j] << ", ";

}

for(n=0;n<60;n++)

{

for (i = 0; i < 7; i++)

{

for (j = 0; j < 1; j++)

{

T[i][j] = 0;

for (k = 0; k <7; k++)

{

T[i][j] +=(A[i][k] \* C1[k][j]);

// T[i][j] += ((0.8\*A[i][k])+0.03) \* C1[k][j];

}

}

}

cout << "\nProduct of matrices ";

for (i = 0; i < 7; i++)

{

for (j = 0; j < 1; j++)

{

cout << T[i][j] << ", ";

C1[i][j]=T[i][j];

}

}

}

return 0;

}

**13. Observations**

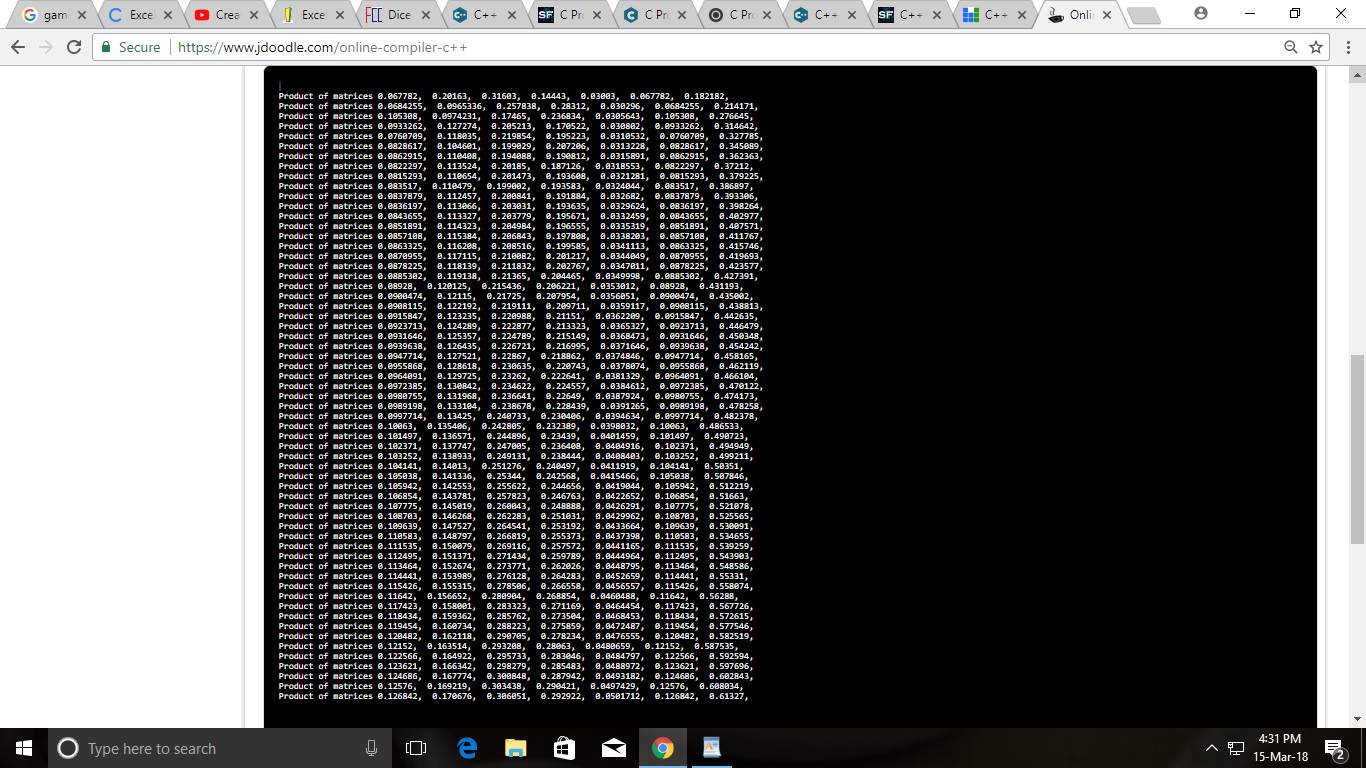
1. The pagerank calculation without teleportation will cause the dead ends have maximum pagerank,

2. In Pagerank calculation with teleportation factor, the values don’t converge to one rather they are reflecting actual nature of the page.

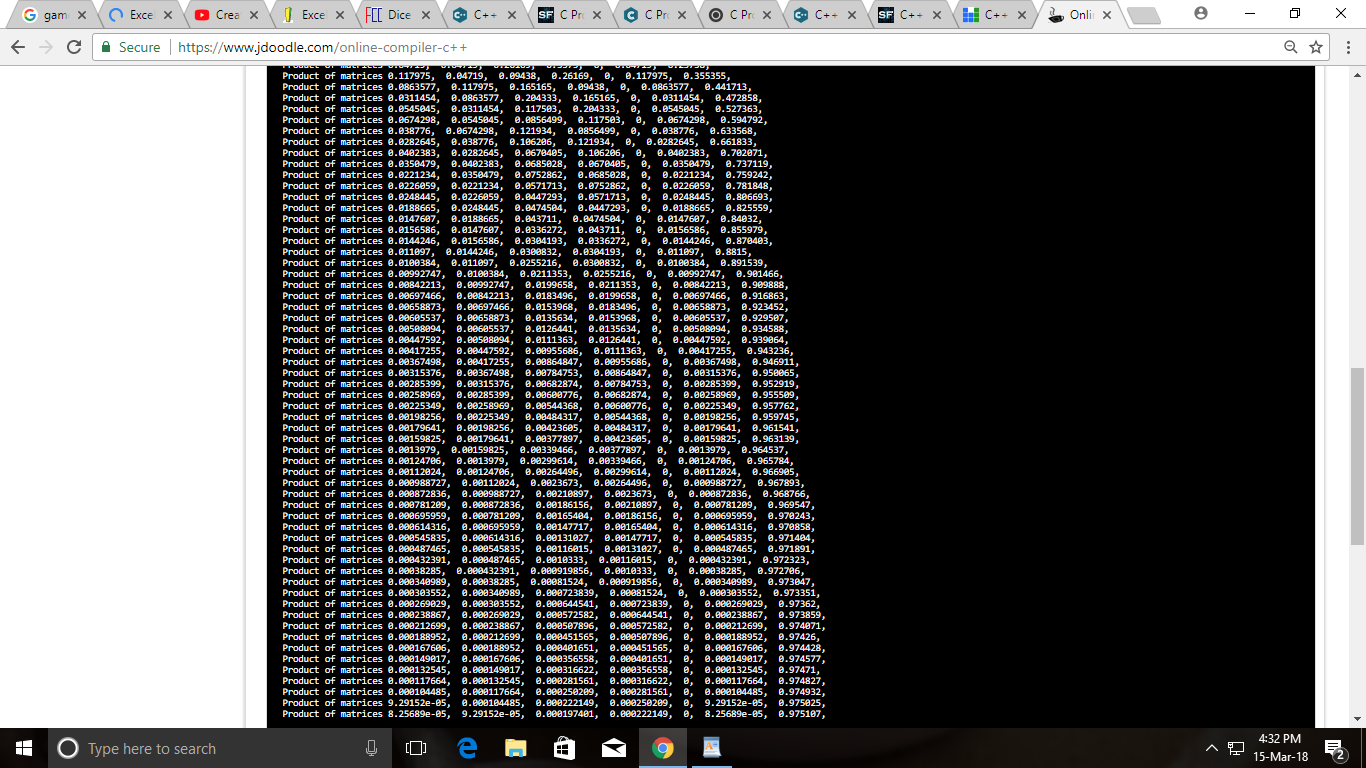
3. Total 100 Iterations were performed.

**14. Results:**

**With Teleportation**



**Without Teleportatoion**



**17. References** :

[1]"Google Press Center: Fun Facts". www.google.com. Archived from the original on 2001-07-15.

[2]"Facts about Google and Competition". Archived from the original on 4 November 2011. Retrieved 12 July 2014.

[3] Sullivan, Danny. "What Is Google PageRank? A Guide For Searchers & Webmasters". Search Engine Land. Archived from the original on 2016-07-03.

[4]Brin, S.; Page, L. (1998). "The anatomy of a large-scale hypertextual Web search engine" (PDF). Computer Networks and ISDN Systems. 30: 107–117. doi:10.1016/S0169-7552(98)00110-X. ISSN 0169-7552. Archived from the original on 2015-09-27.

[5]Gyöngyi, Zoltán; Berkhin, Pavel; Garcia-Molina, Hector; Pedersen, Jan (2006), "Link spam detection based on mass estimation", Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06, Seoul, Korea) (PDF), pp. 439–450, archived (PDF) from the original on 2014-12-03.