NAME: Meith Navlakha
SAP: 60004190068

EXPERIMENT 8

AIM: To implement PageRank Algorithm

THEORY:

PageRank (PR) is an algorithm used by Google Search to rank web pages in their search engine results. PageRanks is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. The numerical weight that it assigns to any given element E is referred to as the PageRank of E and denoted by PR(E).

The algorithm then outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening. Hence, a document with a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to said document.

Damping Factor:

The PageRank theory holds that an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor d. Various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85.

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

Limitations:

- The ordering does not favor current events. According to the algorithm, old pages typically have more votes because they have more links from other reputable pages. Thus, a new page will not be as reputable until it has gained exposure and links from other pages.
- It is prone to manipulation through link selling. Earlier, a market emerged for link selling as Search Engine Optimizers (SEOs) using a way to manipulate the algorithm by creating more traffic to their pages.
- It was prone to link spamming which refers to the practice of leaving links to a page unnecessarily on various platforms.

Other link-based ranking algorithms for Web pages:

- HITS algorithm invented by Jon Kleinberg
- IBM CLEVER project,
- TrustRank algorithm
- Hummingbird algorithm.

CODE:

```
import java.util.*;
import java.util.Arrays;
import java.lang.Integer;
import java.lang.Math;

public class PageRank{
   public static int MAX_NODES = 5;
   public static int[][] incoming = new int[MAX_NODES][MAX_NODES] ;
   public static int[][] outgoing = new int[MAX_NODES][MAX_NODES] ;
   public static float[] PR = new float[MAX_NODES] ;

   public static float calcPR(int curNode , int nodes, int[] Nq){
      float val=0.0f;

      for(int i=0; i<nodes;i++){
         if(incoming[curNode][i]==1){
            val += (float)(PR[i])/(Nq[i]) ;
         }
      }
```

```
    return val;
}

public static void calcPageRank(int nodes, int[] Nq ){
    float d = 0.85f;
    int MAX_ITER = 10;

    for(int i=0; i<MAX_ITER; i++){

        for(int n=0; n<nodes; n++){
            PR[n] = (1-d) + d*(calcPR(n, nodes, Nq));
        }

    }

   System.out.print("\n\n ******* PAGE RANK ******\n") ;
   for(int i =0; i<nodes;i++){
        System.out.print("\n"+(char)(i+65)+" -> "+ PR[i]) ;

   }
}

public static void main(String args[]){

    Scanner sc = new Scanner(System.in) ;

    System.out.print("\n ENTER NUMBER OF NODES :");
    int nodes = sc.nextInt();

    System.out.print("\n ENTER INFLOW MATRIX :\n");
    for(int i=0; i<nodes;i++){
        for(int j=0; j<nodes; j++){
            incoming[i][j] = sc.nextInt();

        }
    }
    int nq=0;
    int[] Nq= new int[nodes];
    System.out.print("\n ENTER OUTFLOW MATRIX :\n");
```

```
        for(int i=0; i<nodes;i++){
            for(int j=0; j<nodes; j++){
                outgoing[i][j] = sc.nextInt();

                if(outgoing[i][j] ==1){
                    nq++;
                }
            }

            Nq[i]= nq;
            nq=0;
        }

        calcPageRank(nodes ,Nq);

        sc.close();
    }
}
```

OUTPUT:

```
C:\Users\meith\OneDrive\Desktop\SEM 5\DWM>java PageRank

 ENTER NUMBER OF NODES :3

 ENTER INFLOW MATRIX :
0 0 1
1 0 0
1 1 0

 ENTER OUTFLOW MATRIX :
0 1 1
0 0 1
1 0 0


 ******* PAGE RANK ******

A -> 1.0769229
B -> 0.7692307
C -> 1.153846
```

```
 ENTER NUMBER OF NODES :4

 ENTER INFLOW MATRIX :
0 0 1 0
1 0 0 0
1 1 0 1
0 0 0 0

 ENTER OUTFLOW MATRIX :
0 1 1 0
0 0 1 0
1 0 0 0
0 0 1 0


 ******* PAGE RANK ******

A -> 1.450153
B -> 0.76631504
C -> 1.5451828
D -> 0.14999998
```

CONCLUSION: In this experiment, I implemented PageRank Algorithm for the given graph. In this experiment I also learnt about the Rank Sink problem and resolved it by using damping factor. In the above example considering a damping factor of 0.85, page C shows the greatest rank followed by D then A and finally B. Another limitation of the PageRank Algorithm is that the rank depends upon the number of Backlinks and not upon the quality of content so, thus during a search, more linked pages are recommended over quality pages.