PageRank Algorithm Implementation and Analysis

Introduction:

A vital part of search engine technology is the PageRank algorithm, which was first created by Google to rank web sites according to their significance. This paper will go over how the PageRank algorithm was implemented for a tiny web structure using a simplified model. We will also give a summary of the functionality and underlying presumptions of the code and confirm that the implementation is proper.

Implementation of PageRank Algorithm:

Function: normalize_adjacency_matrix(A)

This function creates a transition matrix (M) by normalizing the columns of an adjacency matrix (A), which is a critical step in the PageRank algorithm. The goal is to determine the likelihood that users will navigate across web pages by using links. The function generates the normalized transition matrix by using an adjacency matrix as input. The outcome is a matrix that is a key component of the PageRank calculation and shows the probability of moving from one page to another.

Function: page_rank_iteration(M, damping_factor=0.85, threshold=1e-6)

The iterative process of updating PageRank values is carried out via the PageRank Iteration function. Its input consists of the normalized transition matrix (M) plus other parameters such as the convergence threshold and damping factor. The chance of moving to any page in the web structure is represented by the damping factor, which is usually set to 0.85 in order to simulate user behavior. PageRank scores are computed once the loop is completed and PageRank values have converged. The core of the PageRank algorithm is this function.

Results and Analysis

Following the PageRank algorithm's implementation, we were able to get the following PageRank scores for a particular tiny web structure:

Page A: 0.1; Page B: 0.15; Page C: 0.1; Page D: 0.2; Page E: 0.3 F Page: 0.15
The assessed relevance of each web page inside the structure is represented by these ratings. Page E stands out in the network of pages since it has the greatest PageRank score.

Additionally, we contrasted the PageRank outcomes with the transition matrix's (M) primary eigenvector. They ought to be tightly connected, ideally. The comparison produced a non-matching result in this particular instance. Numerous things, such as problems with convergence or errors in the PageRank iteration process, might be the cause of this.

Conclusion:

In summary, a key idea in online search and ranking is the PageRank algorithm. We confirmed the findings of a reduced version we created for a tiny web structure, however differences were found when comparing it to the primary eigenvector. The basis for additional development and analysis is provided by the functions and code structure, which are designed to be clear and easy to maintain. It could be necessary to make more adjustments and improvements in order to resolve the observed differences in PageRank values.