Name : Devkumar Biswas

Class: BE(AI&DS)

Div: B

Subject : IR(CL-II Lab)

Roll no. : BEAD21267

**Assignment No. - 5**

**Problem Statement** :   
Implement Page Rank Algorithm. (Use python or beautiful soup for implementation).

Code:-

pip install beautifulsoup4 requests

import requests

from bs4 import BeautifulSoup

import numpy as np

# Step 1: Scrape a small set of web pages

def get\_links(url):

links = []

try:

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

for link in soup.find\_all('a', href=True):

links.append(link['href'])

except Exception as e:

print(f"Error fetching {url}: {e}")

return links

# Step 2: Create a small web graph

urls = [

'https://www.wikipedia.org/',

'https://en.wikipedia.org/wiki/Web\_scraping',

'https://en.wikipedia.org/wiki/Pagerank',

'https://en.wikipedia.org/wiki/Algorithm',

]

link\_graph = {url: get\_links(url) for url in urls}

# Step 3: Build the adjacency matrix

n = len(urls)

adjacency\_matrix = np.zeros((n, n))

url\_index = {url: i for i, url in enumerate(urls)}

for url, links in link\_graph.items():

for link in links:

if link in url\_index:

adjacency\_matrix[url\_index[url]][url\_index[link]] = 1

# Step 4: Calculate PageRank

def calculate\_pagerank(adjacency\_matrix, d=0.85, num\_iterations=100):

n = adjacency\_matrix.shape[0]

pagerank = np.ones(n) / n

for \_ in range(num\_iterations):

new\_pagerank = (1 - d) / n + d \* (adjacency\_matrix.T @ pagerank) / np.sum(adjacency\_matrix.T @ pagerank)

pagerank = new\_pagerank

return pagerank

pagerank\_values = calculate\_pagerank(adjacency\_matrix)

pagerank\_values\_dict = {url: pagerank\_values[i] for i, url in enumerate(urls)}

# Step 5: Output the results

print("PageRank Values:")

for url, rank in pagerank\_values\_dict.items():

print(f"{url}: {rank:.4f}")

OUTPUT:-

