|  |  |
| --- | --- |
| Name: | Prerna Sunil Jadhav |
| Sap Id: | 60004220127 |
| Class: | T. Y. B. Tech (Computer Engineering) |
| Course: | Big Data Infrastructure Laboratory |
| Course Code: | DJ19CEEL6011 |
| Experiment No.: | 06 |

**AIM:** Implement Matrix Multiplication and Word Frequency Count using Map Reduce

MATRIX MULTIPLICATION USING MAP REDUCE.

**CODE:**

def matrix\_multiply\_mapper(matrix\_a, matrix\_b):

    result = []

    for i in range(len(matrix\_a)):

        for j in range(len(matrix\_b[0])):

            for k in range(len(matrix\_b)):

                result.append(((i, j), (matrix\_a[i][k] \* matrix\_b[k][j])))

    return result

def matrix\_multiply\_reducer(mapped\_result):

    intermediate\_result = {}

    for key, value in mapped\_result:

        if key in intermediate\_result:

            intermediate\_result[key].append(value)

        else:

            intermediate\_result[key] = [value]

    final\_result = []

    for key, values in intermediate\_result.items():

        total = sum(values)

        final\_result.append((key, total))

    return final\_result

def matrix\_multiply(matrix\_a, matrix\_b):

    mapped\_result = matrix\_multiply\_mapper(matrix\_a, matrix\_b)

    reduced\_result = matrix\_multiply\_reducer(mapped\_result)

    final\_result = [[0 for \_ in range(len(matrix\_b[0]))] for \_ in range(len(matrix\_a))]

    for key, value in reduced\_result:

        i, j = key

        final\_result[i][j] = value

    return final\_result

# Example usage:

matrix\_a = [[1, 2],

            [3, 4]]

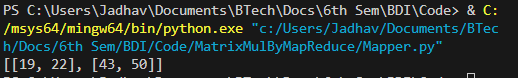
matrix\_b = [[5, 6],

            [7, 8]]

result = matrix\_multiply(matrix\_a, matrix\_b)

print(result)

**OUTPUT:**

****

WORD FREQUENCY COUNT USING MAP REDUCE.

**CODE:**

import re

def word\_count\_mapper(document):

    words = re.findall(r'\w+', document.lower())

    word\_count = {}

    for word in words:

        word\_count[word] = word\_count.get(word, 0) + 1

    return list(word\_count.items())

def word\_count\_reducer(mapped\_result):

    intermediate\_result = {}

    for item in mapped\_result:

        for key, value in item:

            if key in intermediate\_result:

                intermediate\_result[key] += value

            else:

                intermediate\_result[key] = value

    return list(intermediate\_result.items())

def word\_frequency\_count(documents):

    mapped\_result = [word\_count\_mapper(doc) for doc in documents]

    reduced\_result = word\_count\_reducer(mapped\_result)

    final\_result = {}

    for key, value in reduced\_result:

        final\_result[key] = value

    return final\_result

# Example usage:

documents = [

    "This is the first document.",

    "This document is the second document.",

    "And this is the third one.",

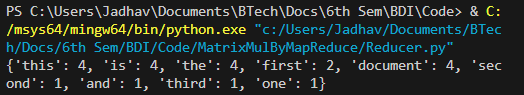
    "Is this the first document?"

]

result = word\_frequency\_count(documents)

print(result)

**OUTPUT:**

****