LP3 (DAA) Lab Exp No.6

import numpy as np

import threading

import time

def matrix\_multiply(A, B):

m, n = len(A), len(B)

p = len(B[0])

result = [[0] \* p for \_ in range(m)]

for i in range(m):

for j in range(p):

for k in range(n):

result[i][j] += A[i][k] \* B[k][j]

return result

def matrix\_multiply\_row(A, B):

m, n = len(A), len(B)

p = len(B[0])

result = [[0] \* p for \_ in range(m)]

def compute\_row(row\_index):

for j in range(p):

for k in range(n):

result[row\_index][j] += A[row\_index][k] \* B[k][j]

threads = [threading.Thread(target=compute\_row, args=(i,)) for i in range(m)]

for thread in threads:

thread.start()

for thread in threads:

thread.join()

return result

def matrix\_multiply\_cell(A, B):

m, n = len(A), len(B)

p = len(B[0])

result = [[0] \* p for \_ in range(m)]

def compute\_cell(row\_index, col\_index):

sum = 0

for k in range(n):

sum += A[row\_index][k] \* B[k][col\_index]

result[row\_index][col\_index] = sum

threads = [threading.Thread(target=compute\_cell, args=(i, j)) for i in range(m) for j in range(p)]

for thread in threads:

thread.start()

for thread in threads:

thread.join()

return result

def print\_matrix(matrix):

for row in matrix:

print(' '.join(map(str, row)))

def main():

while True:

print("Menu: ")

print("1. Single-threaded matrix multiplication")

print("2. Multithreaded (one thread per row)")

print("3. Multithreaded (one thread per cell)")

print("4. Exit")

option = int(input("Choose an option: "))

if option == 4:

break

m = int(input("Enter number of rows for Matrix A: "))

n = int(input("Enter number of columns for Matrix A (and rows for Matrix B): "))

p = int(input("Enter number of columns for Matrix B: "))

A = []

print("Enter matrix A: ")

for \_ in range(m):

A.append(list(map(int, input().split())))

B = []

print("Enter matrix B: ")

for \_ in range(n):

B.append(list(map(int, input().split())))

start\_time = time.time()

if option == 1:

result = matrix\_multiply(A, B)

print("Single-threaded result: ")

elif option == 2:

result = matrix\_multiply\_row(A, B)

print("Multithreaded (one thread per row) result: ")

elif option == 3:

result = matrix\_multiply\_cell(A, B)

print("Multithreaded (one thread per cell) result: ")

else:

print("Invalid option")

continue

end\_time = time.time()

print\_matrix(result)

print(f"Time: {(end\_time - start\_time) \* 1000:.2f} ms")

if \_\_name\_\_ == "\_\_main\_\_":

main()

OUTPUT: -





