Prompting Chatgpt as follows:

* Can you write for me matrix multiplication in java and python languages that works with any sizes?

Answer:

import java.util.ArrayList;

import java.util.Arrays;

public class MatrixMultiplication {

public static void main(String[] args) {

// Example matrices using ArrayList

ArrayList<ArrayList<Integer>> A = new ArrayList<>();

A.add(new ArrayList<>(Arrays.asList(1, 2, 3)));

A.add(new ArrayList<>(Arrays.asList(4, 5, 6)));

ArrayList<ArrayList<Integer>> B = new ArrayList<>();

B.add(new ArrayList<>(Arrays.asList(7, 8)));

B.add(new ArrayList<>(Arrays.asList(9, 10)));

B.add(new ArrayList<>(Arrays.asList(11, 12)));

ArrayList<ArrayList<Integer>> result = multiplyMatrices(A, B);

// Print the result

System.out.println("Matrix Multiplication Result:");

for (ArrayList<Integer> row : result) {

System.out.println(row);

}

}

public static ArrayList<ArrayList<Integer>> multiplyMatrices(ArrayList<ArrayList<Integer>> A, ArrayList<ArrayList<Integer>> B) {

int rowsA = A.size();

int colsA = A.get(0).size();

int colsB = B.get(0).size();

// Initialize the result matrix

ArrayList<ArrayList<Integer>> result = new ArrayList<>();

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

result.add(new ArrayList<>(Arrays.asList(new Integer[colsB])));

Arrays.fill(result.get(i).toArray(), 0);

}

// Matrix multiplication using nested loops

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

for (int j = 0; j < colsB; j++) {

for (int k = 0; k < colsA; k++) {

result.get(i).set(j, result.get(i).get(j) + A.get(i).get(k) \* B.get(k).get(j));

}

}

}

return result;

}

}

def calculate\_matrix\_product(mat\_a, mat\_b):

# Get the dimensions of the input matrices

rows\_a = len(mat\_a)

cols\_a = len(mat\_a[0])

rows\_b = len(mat\_b)

cols\_b = len(mat\_b[0])

# Check if the matrices can be multiplied

if cols\_a != rows\_b:

raise ValueError("Cannot multiply: number of columns in A must equal number of rows in B.")

product\_matrix = [

[

sum(mat\_a[i][k] \* mat\_b[k][j] for k in range(cols\_a))

for j in range(cols\_b)

]

for i in range(rows\_a)

]

return product\_matrix

# Example matrices of any sizes

matrix\_a = [

[1, 2, 3],

[4, 5, 6]

]

matrix\_b = [

[7, 8],

[9, 10],

[11, 12]

]

# Calculate the product of the matrices

result\_product = calculate\_matrix\_product(matrix\_a, matrix\_b)

# Print the result

print("Result of A \* B:")

for row in result\_product:

print(row)

* Can you do more efficient ones?

In Python

import numpy as np

def efficient\_matrix\_multiplication(mat\_a, mat\_b):

# Convert lists to NumPy arrays

np\_a = np.array(mat\_a)

np\_b = np.array(mat\_b)

# Use NumPy's dot function for matrix multiplication

product\_matrix = np.dot(np\_a, np\_b)

return product\_matrix

# Example matrices of any sizes

matrix\_a = [

[1, 2, 3],

[4, 5, 6]

]

matrix\_b = [

[7, 8],

[9, 10],

[11, 12]

]

# Calculate the product of the matrices

result\_product = efficient\_matrix\_multiplication(matrix\_a, matrix\_b)

# Print the result

print("Result of A \* B:")

print(result\_product)

In java

public class OptimizedMatrixMultiplier {

public static int[][] multiplyMatrices(int[][] matrixA, int[][] matrixB) {

int rowsA = matrixA.length;

int colsA = matrixA[0].length;

int rowsB = matrixB.length;

int colsB = matrixB[0].length;

// Check if multiplication is possible

if (colsA != rowsB) {

throw new IllegalArgumentException("Cannot multiply: number of columns in A must equal number of rows in B.");

}

int[][] productMatrix = new int[rowsA][colsB];

// Matrix multiplication with loop unrolling

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

for (int j = 0; j < colsB; j++) {

int sum = 0;

int k = 0;

for (; k < colsA - 4; k += 4) {

sum += matrixA[i][k] \* matrixB[k][j];

sum += matrixA[i][k + 1] \* matrixB[k + 1][j];

sum += matrixA[i][k + 2] \* matrixB[k + 2][j];

sum += matrixA[i][k + 3] \* matrixB[k + 3][j];

}

// Handle remaining elements

for (; k < colsA; k++) {

sum += matrixA[i][k] \* matrixB[k][j];

}

productMatrix[i][j] = sum;

}

}

return productMatrix;

}

public static void main(String[] args) {

// Example matrices

int[][] matrixA = {

{1, 2, 3, 4},

{5, 6, 7, 8}

};

int[][] matrixB = {

{9, 10},

{11, 12},

{13, 14},

{15, 16}

};

// Calculate the product of the matrices

int[][] resultProduct = multiplyMatrices(matrixA, matrixB);

// Print the result

System.out.println("Result of A \* B (Optimized):");

for (int[] row : resultProduct) {

for (int value : row) {

System.out.print(value + " ");

}

System.out.println();

}

}

}