PCDIT Worksheet

(P)roblem Statement:

From problem description, identify the input and the output, and summarize the process to get the output from the input.

Input: matrix A , matrix B } list of list
$p \times q$ $q \times r$

Output: matrix C
$p \times r$

Process/Computation: Matrix Multiplication

Test (C)ases:

Create test cases. Generate different kinds of input and work out the outputs. Think of different possible inputs and outputs.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times B = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

A = [[1,2,3], [4,5,6]]
B = [[1,4], [2,5] [3,6]]

$$C = \begin{bmatrix} 1\times1 + 2\times2 + 3\times3 & 1\times4 + 2\times5 + 3\times6 \\ 4\times1 + 5\times2 + 6\times3 & 4\times4 + 5\times5 + 6\times6 \end{bmatrix} = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}$$

c = [[14, 32], [32, 77]]

$$C = \begin{bmatrix} a[0][0]\,b[0][0] + a[0][1]\,b[1][0] + a[0][2]\,b[2][0] & a[0][0]\,b[0][1] + a[0][1]\,b[1][1] + a[0][2]\,b[2][1] \\ a[1][0]\,b[0][0] + a[1][1]\,b[1][0] + a[1][2]\,b[2][0] & a[1][0]\,b[0][1] + a[1][1]\,b[1][1] + a[1][2]\,b[2][1] \end{bmatrix}$$

(D)esign Algorithm:

From test cases, write down the steps in English. Identify sequential, repetition, and conditionals structures, and revise the steps. Use certain keywords such as: if … then ..., while … do ..., for every … do .... Generate pseudocode. Revise several times.

Get # of rows for A & B
Get # of cols for A & B

- For each element in the first row of A; multiply w/ the respective element in the first col of B, and sum them up.
  Put the sum in the first element of the first row of C.
- For each element in the first row of A, multiply w/ the respective element in the second col of B, and sum them up.
  Put the sum in the second element of the first row of C.
  do this for every col in B.
- For each element in the second row of A, multiply w/ the respective element in the first column of B, and sum them up.
  Put the sum in the ~~second~~ first element of the second row of C
- For each element in the second row of A, multiply w/ the respective element in the second col of B, and sum them up
  Put the sum in the second element of the second row of C.
  do this for every col in B.

do this for every row in A.

(I)mplementation:

Implement the pseudocode in Python

(T)esting:

Test the Python code using different test cases. Use a debugger.

(D)esign Algorithm:

From test cases, write down the steps in English. Identify sequential, repetition, and conditionals structures, and revise the steps. Use certain keywords such as: if … then ..., while … do ..., for every … do .... Generate pseudocode. Revise several times.

1. Get # of rows for A & B
2. Get # of cols for A & B
3. For the number of rows in A, do:
   a. create empty list for current row
   b. For the number of cols in B, do:
      a. initialize sum to zero
      b. For the number of cols in A or number of rows in B, do:
         a. multiply the respective col in A with the respective row in B.
         b. Add this to sum.
      c. Add sum to the list of current row
   c. Add list of current row to the final list
4. Return final list

```
def matrixMul(A, B):
    rowA = len(A)
    rowB = len(B)
    colA = len(A[0])
    colB = len(B[0])

    for i in range(rowA):
        curRow = []
        for j in range(colB):
            sum = 0
            for k in range(colA):
                sum += A[i][k] * B[k][j]
            curRow.append(sum)
        finalList.append(curRow)
    return finalList
```

(I)mplementation:

Implement the pseudocode in Python

(T)esting:

Test the Python code using different test cases. Use a debugger.

PCDIT Worksheet

(P)roblem Statement:

From problem description, identify the input and the output, and summarize the process to get the output from the input.

Input:

Output:

Process/Computation:

Test (C)ases:

Create test cases. Generate different kinds of input and work out the outputs. Think of different possible inputs and outputs.