LAB ASSIGNMENT 1

Name: Atharva Kangralkar

PRN: 12311493

Class: TY

Batch: 2

Aim: Analysis of Matrix Multiplication.

Step 1: Pseudocode for Matrix Multiplication

```
start

declare integers r1, c1, r2, c2, i, j, k
declare 2d arrays a, b, c

print "enter number of rows and columns for matrix a: "
input r1, c1

print "enter number of rows and columns for matrix b: "
input r2, c2

if c1 != r2 then
   print "error! the number of columns in a must be equal to the number of rows in b."
   exit program
end if

initialize matrix a with dimensions r1 x c1
initialize matrix b with dimensions r2 x c2
initialize matrix c with dimensions r1 x c2 and fill with 0

print "enter elements of matrix a:"
for i from 0 to r1-1 do
   for j from 0 to c1-1 do
      input a[i][j]
   end for
end for

print "enter elements of matrix b:"
```

```
for i from 0 to r2-1 do
   for j from 0 to c2-1 do
      input b[i][j]
   end for
end for

for i from 0 to r1-1 do
   for j from 0 to c2-1 do
      set c[i][j] = 0
      for k from 0 to c1-1 do
         c[i][j] = c[i][j] + (a[i][k] * b[k][j])
      end for
   end for
end for

print "resultant matrix after multiplication:"
for i from 0 to r1-1 do
   for j from 0 to c2-1 do
      print c[i][j], " "
   end for
   print newline
end for

end
```

Step 2: Code for Matrix Multiplication

```
#include <bits/stdc++.h>

using namespace std;


int main() {

   int r1, c1, r2, c2, i, j, k;


   cout << "Enter number of rows and columns for matrix A: ";

   cin >> r1 >> c1;

   cout << "Enter number of rows and columns for matrix B: ";

   cin >> r2 >> c2;


   if (c1 != r2) {
```

```
    cout << "Error! The number of columns in A must be equal to the number of rows in B."
<< endl;
    return 1;
  }


  vector<vector<int>> A;
  vector<vector<int>> B;
  vector<vector<int>> C;

  cout << "Enter elements of matrix A:" << endl;
  for (i = 0; i < r1; i++) {
    for (j = 0; j < c1; j++) {
      cin >> A[i][j];
    }
  }

  cout << "Enter elements of matrix B:" << endl;
  for (i = 0; i < r2; i++) {
    for (j = 0; j < c2; j++) {
      cin >> B[i][j];
    }
  }

  for (i = 0; i < r1; i++) {
    for (j = 0; j < c2; j++) {
      for (k = 0; k < c1; k++) {
        C[i][j] += A[i][k] * B[k][j];
      }
    }
```

```
    }

    cout << "Resultant matrix after multiplication:" << endl;
    for (i = 0; i < r1; i++) {
        for (j = 0; j < c2; j++) {
            cout << C[i][j] << "\t";
        }
        cout << endl;
    }

    return 0;
}
```

Step 3: Equations for number of arithmetic operations needed in matrix multiplication. (Considering addition and Multiplication only).

To compute each of the m×p elements in the resulting matrix, a dot product is performed. This dot product involves n multiplications and n−1 additions.

- Number of Multiplications: The total number of multiplications is the number of elements in the final matrix multiplied by the number of multiplications per element.

$M=(m\times p)\times n$

- Number of Additions: Similarly, the total number of additions is:

$A=(m\times p)\times(n-1)$

Two Square Matrices (n×n)

Here m=n=p.

- **Number of Multiplications**:

$M(n)=n\times n\times n=n^3$

- **Number of Additions**:

A(n)=n×n×(n−1)=n^3−n^2

- **Total Arithmetic Operations**:

T(n)=M(n)+A(n)=n^3+(n^3−n^2)=2n^3−n^2

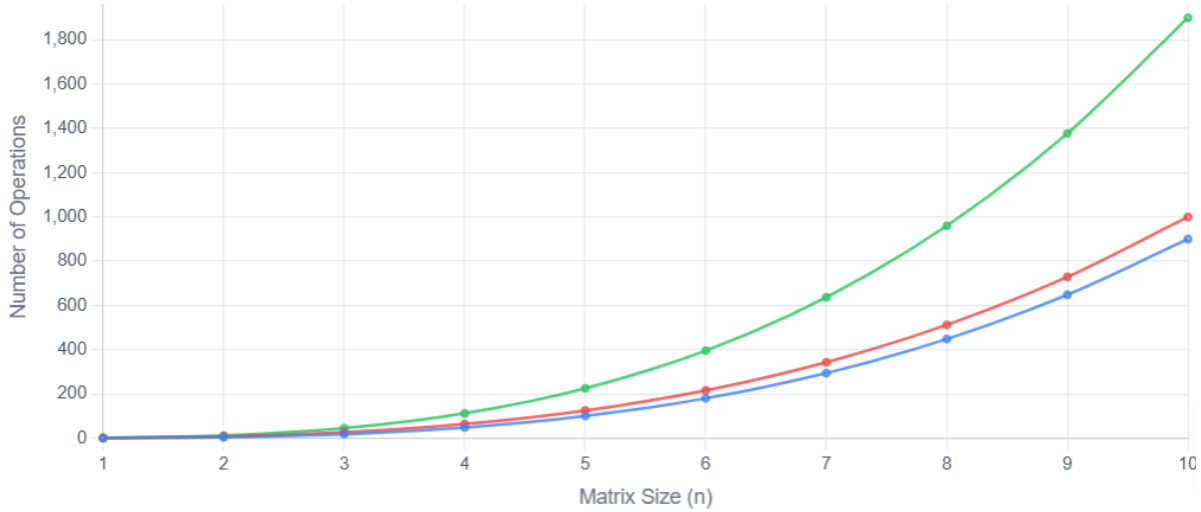
simplified to T(n)=2n^3

So total Time Complexity = n^3

Step 4: Prepare a table for at least for 10 values of n.

n	**A ($n^3 - n^2$)**	**M (n^3)**	**T ($2n^3 - n^2$)**
1	0	1	7
2	4	8	28
3	18	27	63
4	48	64	112
5	100	125	175
6	180	216	252
7	294	343	343
8	448	512	448
9	648	729	567
10	900	1000	700

Step 5: Comparing n versus addition, Multiplication and Total operation in Matrix multiplication.



Output:

//test output

1)

```
Enter number of rows and columns for matrix A: 2 3
Enter number of rows and columns for matrix B: 3 2
Enter elements of matrix A:
1 2 3
4 5 6
Enter elements of matrix B:
7 8
9 10
11 12
Resultant matrix after multiplication:
58  64
139 154
```

2)

```
Enter number of rows and columns for matrix A: 2 3
Enter number of rows and columns for matrix B: 2 2
Error! The number of columns in A must be equal to the number of rows in B.
```

3)

```
Enter number of rows and columns for matrix A: 1 3
Enter number of rows and columns for matrix B: 3 1
Enter elements of matrix A:
1 2 3
Enter elements of matrix B:
4
5
6
Resultant matrix after multiplication:
32
```