**Experiment 5**

**Aim:** To implement Strassen’s matrix multiplication.

**Code:**

import random as rn

def add\_mat(A,B,N):

S = [[0 for i in range(N)] for i in range(N)]

for i in range(N):

for j in range(N):

S[i][j] = A[i][j] + B[i][j]

return S

def sub\_mat(A,B,N):

S = [[0 for i in range(N)] for i in range(N)]

for i in range(N):

for j in range(N):

S[i][j] = A[i][j] - B[i][j]

return S

def s\_multiplication(A,B,N):

C = [[0 for i in range(N)] for i in range(N)]

if(N == 1):

return A[0][0]\*B[0][0]

else:

mid = N//2

A11 = [[A[i][j] for j in range(0,mid)] for i in range(0,mid)]

A12 = [[A[i][j] for j in range(mid,N)] for i in range(0,mid)]

A21 = [[A[i][j] for j in range(0,mid)] for i in range(mid,N)]

A22 = [[A[i][j] for j in range(mid,N)] for i in range(mid,N)]

B11 = [[B[i][j] for j in range(0,mid)] for i in range(0,mid)]

B12 = [[B[i][j] for j in range(mid,N)] for i in range(0,mid)]

B21 = [[B[i][j] for j in range(0,mid)] for i in range(mid,N)]

B22 = [[B[i][j] for j in range(mid,N)] for i in range(mid,N)]

S1 = sub\_mat(B12,B22,mid)

S2 = add\_mat(A11,A12,mid)

S3 = add\_mat(A21,A22,mid)

S4 = sub\_mat(B21,B11,mid)

S5 = add\_mat(A11,A22,mid)

S6 = add\_mat(B11,B22,mid)

S7 = sub\_mat(A12,A22,mid)

S8 = add\_mat(B21,B22,mid)

S9 = sub\_mat(A11,A21,mid)

S10 = add\_mat(B11,B12,mid)

P1 = s\_multiplication(A11,S1,mid)

P2 = s\_multiplication(S2,B22,mid)

P3 = s\_multiplication(S3,B11,mid)

P4 = s\_multiplication(A22,S4,mid)

P5 = s\_multiplication(S5,S6,mid)

P6 = s\_multiplication(S7,S8,mid)

P7 = s\_multiplication(S9,S10,mid)

#print(P1,P2,P3,P4,P5,P6,P7)

if(mid == 1):

C[0][0] = P5 + P4 - P2 + P6

C[0][1] = P1 + P2

C[1][0] = P3 + P4

C[1][1] = P5 + P1 - P3 - P7

#print(C)

return C

C11 = [[0 for j in range(0,mid)] for i in range(0,mid)]

C12 = [[0 for j in range(0,mid)] for i in range(0,mid)]

C21 = [[0 for j in range(0,mid)] for i in range(0,mid)]

C22 = [[0 for j in range(0,mid)] for i in range(0,mid)]

for i in range(0,mid):

for j in range(0,mid):

C11[i][j] = P5[i][j] + P4[i][j] - P2[i][j] + P6[i][j]

C12[i][j] = P1[i][j] + P2[i][j]

C21[i][j] = P3[i][j] + P4[i][j]

C22[i][j] = P5[i][j] + P1[i][j] - P3[i][j] - P7[i][j]

for i in range(0,mid):

for j in range(0,mid):

C[i][j] = C11[i][j]

C[i][j+mid] = C12[i][j]

C[i+mid][j] = C21[i][j]

C[i+mid][j+mid] = C22[i][j]

#print(C)

return C

print('Enter the size of the matrix: ')

N = int(input())

A = [[rn.randint(1,9) for j in range(N)] for i in range(N)]

B = [[rn.randint(1,9) for j in range(N)] for i in range(N)]

print('First Matrix: ')

for i in range(N):

for j in range(N):

print(A[i][j], end = " ")

print(end = '\n')

print('Second Matrix: ')

for i in range(N):

for j in range(N):

print(B[i][j], end = " ")

print(end = '\n')

C = s\_multiplication(A,B,N)

print('Resultant Matrix: ')

for i in range(N):

for j in range(N):

print(C[i][j], end = " ")

print(end = '\n')

**Output:**

**Text

Description automatically generated**

**Conclusion:** Strassen’s Matrix Multiplication algorithm has been implemented successfully.