# CSCE 5150 – Analysis of Computer Algorithms

# Programming Assignment 1 – Divide and Conquer

# Neha Goud Baddam

# 11519516

A computer program to implement the Strassen' algorithm for matrix multiplication.

In [1]:

```
import numpy as np

#below is the function defined for strassens algorithm
# we are passing x and y matrices as parameters
def strassen(matrix_A, matrix_B):

    #if the size of the matrix is 1, then just multiply the matrices
    if matrix_A.size == 1 or matrix_B.size == 1:
        return matrix_A * matrix_B

    #firstly we get the size of the matrix
    sizeOfMatrix = matrix_A.shape[0]

    #if the size of the matrix is odd, we pad the matrix with zeros to get a ever
    if sizeOfMatrix % 2 == 1:
        matrix_A  = np.pad(matrix_A, (0, 1), mode='constant')
        matrix_B = np.pad(matrix_B, (0, 1), mode='constant')

    #now we divide the matrix A and matrix B into half and store it in the below
    #get the nearsest interger of size of the matrix / 2
    halfOfSize= int(np.ceil(sizeOfMatrix / 2))

    #partitioning into sub matrices
    A11 = matrix_A[: halfOfSize, : halfOfSize]
    A12 = matrix_A[: halfOfSize, halfOfSize:]
    A21 = matrix_A[halfOfSize:, : halfOfSize]
    A22 = matrix_A[halfOfSize:, halfOfSize:]
    B11 = matrix_B[: halfOfSize, : halfOfSize]
    B12 = matrix_B[: halfOfSize, halfOfSize:]
    B21 = matrix_B[halfOfSize:, : halfOfSize]
    B22 = matrix_B[halfOfSize:, halfOfSize:]

    #finding the sum and differences of the matrices based on the below formulas
    s1 = B12 - B22
    s2 = A11 + A12
    s3 = A21 + A22
    s4 = B21 - B11
    s5 = A11 + A22
    s6 = B11 + B22
    s7 = A12 - A22
    s8 = B21 + B22
    s9 = A11 - A21
    s10= B11 + B12

    #finding the product of the matrices using the below formula
    #recurssively call the function strassen to divide the matrix until it cannoy
    #any submatrix

    p1 = strassen(A11, s1)
    p2 = strassen(s2, B22)
    p3 = strassen(s3, B11)
    p4 = strassen(A22, s4)
    p5 = strassen(s5, s6)
    p6 = strassen(s7, s8)
    p7 = strassen(s9, s10)
    #now finally create a result matrix of original size of mtraix A and B and st

    matrix_C = np.zeros((2 * halfOfSize, 2 * halfOfSize), dtype=np.int32)

    matrix_C[: halfOfSize, : halfOfSize] = p5 + p4 - p2 + p6
    matrix_C[: halfOfSize, halfOfSize:] = p1 + p2
    matrix_C[halfOfSize:, : halfOfSize] = p3 + p4
    matrix_C[halfOfSize:, halfOfSize:] = p1 + p5 - p3 - p7

    #now we return the matrix_c value that has the result
    return matrix_C[: sizeOfMatrix, : sizeOfMatrix]
```

In [2]:

```

#take the size of matrix as input
matrixSize = int(input("Enter matrix size for matrix A and matrix B : "))

#create two arrays of size given above
arr1 = np.zeros(shape = (matrixSize,matrixSize))
arr2 = np.zeros(shape = (matrixSize,matrixSize))

#take matrix A and matrix B as input for matrix multiplication
for i in range(0,matrixSize):
    for j in range(0,matrixSize):
        arr1[i][j] = input("Enter numbers in the first matrix A, one after the ot

for i in range(0,matrixSize):
    for j in range(0,matrixSize):
        arr2[i][j] = input("Enter numbers in the second matrix B, one after the o

print("Below is the matrix_A",arr1)
print("Below is the matrix_B",arr2)


matrix_A = np.array(arr1)
matrix_B = np.array(arr2)

print('Strassens Matrix multiplication result for above matrix_A and matrix_B is
print(strassen(matrix_A, matrix_B))
```

```
Enter matrix size for matrix A and matrix B : 2
Enter numbers in the first matrix A, one after the other: 1
Enter numbers in the first matrix A, one after the other: 3
Enter numbers in the first matrix A, one after the other: 7
Enter numbers in the first matrix A, one after the other: 5
Enter numbers in the second matrix B, one after the other: 6
Enter numbers in the second matrix B, one after the other: 8
Enter numbers in the second matrix B, one after the other: 4
Enter numbers in the second matrix B, one after the other: 2
Below is the matrix_A [[1. 3.]
 [7. 5.]]
Below is the matrix_B [[6. 8.]
 [4. 2.]]
Strassens Matrix multiplication result for above matrix_A and matrix_B is :
[[18 14]
 [62 66]]
```

In [3]:

```python
#take the size of matrix as input
matrixSize = int(input("Enter matrix size for matrix A and matrix B : "))

#create two arrays of size given above
arr1 = np.zeros(shape = (matrixSize,matrixSize))
arr2 = np.zeros(shape = (matrixSize,matrixSize))

#take matrix A and matrix B as input for matrix multiplication
for i in range(0,matrixSize):
    for j in range(0,matrixSize):
        arr1[i][j] = input("Enter numbers in the first matrix A, one after the ot

for i in range(0,matrixSize):
    for j in range(0,matrixSize):
        arr2[i][j] = input("Enter numbers in the second matrix B, one after the o

print("Below is the matrix_A",arr1)
print("Below is the matrix_B",arr2)


matrix_A = np.array(arr1)
matrix_B = np.array(arr2)

print('Strassens Matrix multiplication result for above matrix_A and matrix_B is
print(strassen(matrix_A, matrix_B))
```

```
Enter matrix size for matrix A and matrix B : 3
Enter numbers in the first matrix A, one after the other: 2
Enter numbers in the first matrix A, one after the other: 3
Enter numbers in the first matrix A, one after the other: 4
Enter numbers in the first matrix A, one after the other: 5
Enter numbers in the first matrix A, one after the other: 6
Enter numbers in the first matrix A, one after the other: 7
Enter numbers in the first matrix A, one after the other: 8
Enter numbers in the first matrix A, one after the other: 9
Enter numbers in the first matrix A, one after the other: 0
Enter numbers in the second matrix B, one after the other: 12
Enter numbers in the second matrix B, one after the other: 45
Enter numbers in the second matrix B, one after the other: 76
Enter numbers in the second matrix B, one after the other: 8
Enter numbers in the second matrix B, one after the other: 9
Enter numbers in the second matrix B, one after the other: 3
Enter numbers in the second matrix B, one after the other: 4
Enter numbers in the second matrix B, one after the other: 5
Enter numbers in the second matrix B, one after the other: 6
Below is the matrix_A [[2. 3. 4.]
 [5. 6. 7.]
 [8. 9. 0.]]
Below is the matrix_B [[12. 45. 76.]
 [ 8.  9.  3.]
 [ 4.  5.  6.]]
Strassens Matrix multiplication result for above matrix_A and matrix_B is :
[[ 64 137 185]
 [136 314 440]
 [168 441 635]]
```

In [4]:

```
#take the size of matrix as input
matrixSize = int(input("Enter matrix size for matrix A and matrix B : "))

#create two arrays of size given above
arr1 = np.zeros(shape = (matrixSize,matrixSize))
arr2 = np.zeros(shape = (matrixSize,matrixSize))

#take matrix A and matrix B as input for matrix multiplication
for i in range(0,matrixSize):
    for j in range(0,matrixSize):
        arr1[i][j] = input("Enter numbers in the first matrix A, one after the ot

for i in range(0,matrixSize):
    for j in range(0,matrixSize):
        arr2[i][j] = input("Enter numbers in the second matrix B, one after the c

print("Below is the matrix_A",arr1)
print("Below is the matrix_B",arr2)


matrix_A = np.array(arr1)
matrix_B = np.array(arr2)

print('Strassens Matrix multiplication result for above matrix_A and matrix_B is
print(strassen(matrix_A, matrix_B))
```

```
Enter matrix size for matrix A and matrix B : 4
Enter numbers in the first matrix A, one after the other: 1
Enter numbers in the first matrix A, one after the other: 2
Enter numbers in the first matrix A, one after the other: 3
Enter numbers in the first matrix A, one after the other: 4
Enter numbers in the first matrix A, one after the other: 5
Enter numbers in the first matrix A, one after the other: 6
Enter numbers in the first matrix A, one after the other: 7
Enter numbers in the first matrix A, one after the other: 8
Enter numbers in the first matrix A, one after the other: 9
Enter numbers in the first matrix A, one after the other: 1
Enter numbers in the first matrix A, one after the other: 2
Enter numbers in the first matrix A, one after the other: 3
Enter numbers in the first matrix A, one after the other: 4
Enter numbers in the first matrix A, one after the other: 6
Enter numbers in the first matrix A, one after the other: 7
Enter numbers in the first matrix A, one after the other: 8
Enter numbers in the second matrix B, one after the other: 6
Enter numbers in the second matrix B, one after the other: 4
Enter numbers in the second matrix B, one after the other: 3
Enter numbers in the second matrix B, one after the other: 6
Enter numbers in the second matrix B, one after the other: 8
Enter numbers in the second matrix B, one after the other: 4
Enter numbers in the second matrix B, one after the other: 2
Enter numbers in the second matrix B, one after the other: 9
Enter numbers in the second matrix B, one after the other: 5
Enter numbers in the second matrix B, one after the other: 2
Enter numbers in the second matrix B, one after the other: 1
Enter numbers in the second matrix B, one after the other: 4
Enter numbers in the second matrix B, one after the other: 3
Enter numbers in the second matrix B, one after the other: 6
Enter numbers in the second matrix B, one after the other: 8
Enter numbers in the second matrix B, one after the other: 9
Below is the matrix_A [[1. 2. 3. 4.]
 [5. 6. 7. 8.]
 [9. 1. 2. 3.]
 [4. 6. 7. 8.]]
Below is the matrix_B [[6. 4. 3. 6.]
 [8. 4. 2. 9.]
 [5. 2. 1. 4.]
 [3. 6. 8. 9.]]
Strassens Matrix multiplication result for above matrix_A and matrix_B is :
[[ 49  42  42  72]
 [137 106  98 184]
 [ 81  62  55  98]
 [131 102  95 178]]
```

```
In [ ]:
```