

### 83. Strassen's matrix multiplication

Program:

```
import numpy as np
```

```
def strassen_matrix_multiply(A, B):
    if len(A) == 1:
        return A * B

    n = len(A) // 2
    a11 = A[:n, :n]
    a12 = A[:n, n:]
    a21 = A[n:, :n]
    a22 = A[n:, n:]

    b11 = B[:n, :n]
    b12 = B[:n, n:]
    b21 = B[n:, :n]
    b22 = B[n:, n:]

    p1 = strassen_matrix_multiply(a11 + a22, b11 + b22)
    p2 = strassen_matrix_multiply(a21 + a22, b11)
    p3 = strassen_matrix_multiply(a11, b12 - b22)
    p4 = strassen_matrix_multiply(a22, b21 - b11)
    p5 = strassen_matrix_multiply(a11 + a12, b22)
    p6 = strassen_matrix_multiply(a21 - a11, b11 + b12)
    p7 = strassen_matrix_multiply(a12 - a22, b21 + b22)

    c11 = p1 + p4 - p5 + p7
    c12 = p3 + p5
    c21 = p2 + p4
    c22 = p1 - p2 + p3 + p6

    C = np.vstack((np.hstack((c11, c12)), np.hstack((c21, c22))))
    return C
```

# Example Usage

```
A = np.array([[1, 2], [3, 4]])
```

```
B = np.array([[5, 6], [7, 8]])
```

```
result = strassen_matrix_multiply(A, B)
```

```
print(result)
```

Output:

```
[[19 22]
 [43 50]]
```

```
=== Code Execution Successful ===
```

Time complexity:  $O(n^3)$