EXERCISE-83  Strassens matrix multiplication

PROGRAM

```python
import numpy as np
def strassen(matrix1, matrix2):
    if len(matrix1) == 1:
        return matrix1 * matrix2
    a, b, c, d = split(matrix1)
    e, f, g, h = split(matrix2)
    p1 = strassen(a, f - h)
    p2 = strassen(a + b, h)
    p3 = strassen(c + d, e)
    p4 = strassen(d, g - e)
    p5 = strassen(a + d, e + h)
    p6 = strassen(b - d, g + h)
    p7 = strassen(a - c, e + f)
    result_a = p5 + p4 - p2 + p6
    result_b = p1 + p2
    result_c = p3 + p4
    result_d = p1 + p5 - p3 - p7
    result_top = np.concatenate((result_a, result_b), axis=1)
    result_bottom = np.concatenate((result_c, result_d), axis=1)
    result = np.concatenate((result_top, result_bottom), axis=0)
    return result
def split(matrix):
    row, col = matrix.shape
    row2, col2 = row//2, col//2
    return matrix[:row2, :col2], matrix[:row2, col2:], matrix[row2:, :col2], matrix[row2:, col2:]
A = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]])
B = np.array([[17, 18, 19, 20], [21, 22, 23, 24], [25, 26, 27, 28], [29, 30, 31, 32]])
result = strassen(A, B)
print(result)
```

OUTPUT

```
[[ 250  260  270  280]
 [ 618  644  670  696]
 [ 986 1028 1070 1112]
 [1354 1412 1470 1528]]
```

TIME COMPLEXITY O(n^3))