

Assignment Description: Implement Strassen's Matrix Multiplication Algorithm (50 points)

Sample code has been provided that implements a basic implementation of Divide and Conquer for Matrix Multiplication. You should use this code as your base (copy+paste) into your own function that implements Strassen's method explained in Week 5 slides. Your assignment is to implement Strassen's algorithm for Matrix Multiplication. Submit a python file with a function "strassens(A,B)" that can be called and tested.

Submission Instructions Please submit a single python file (.py) with a function defined as described blow. If you would like to include any of your own test code, please include at bottom using:

```
if __name__ == "__main__":
```

This allows the class to be imported for grading/testing without executing your tests. Also, please refrain from including print statements in your submission.

Please name the file:

Strassens.py

Your submission should contain the following function definition:

```
def strassens(A, B):
    """
    Uses Strassen's method for matrix multiplication to return the product of the two
    provided Matrices A and B.
    :param A: the first matrix to multiply. Implemented as a np.array
    :param B: The second matrix to multiply. Implemented as a np.array
    :return: The product of A and B
    """
```

Description of the provided code The file MatrixMult_Starter.py contains two basic methods.

```
def recursive_matrix_mult(A, B):
    """
    Recursively multiplies sub matrices of A and B
    :param a: The first Matrix A implemented as a np.array
    :param b: The second Matrix B implemented as a np.array
    :return: The product of a and b
    """

def basic_smoke_test():
    """ Calls recursive_matrix_mult and throws AssertionError if calculations are not
    performed as expected.
    """
```