DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
Algorithm Laboratory (CSLR41)

Assignment 7

Problem Statement: Given a set of matrices, find the optimal way of multiplying these matrices such that the total number of multiplications is minimum. Then find the multiplied matrix.

Part A: Compare the classical recursion based method with dynamic programming based method to find the optimal number of multiplication to multiply a given chain of matrices.

Input: A chain of n number of matrices which can be multiplied.

Part B: Find the final multiplied matrix

1. When all matrices are square matrices: implement Strassen's matrix multiplication
2. Otherwise write an efficient algorithm to multiply the matrices as per the optimal chain given by Part A.

Input: The actual n matrices whose orders are given in Part A.

Tasks:

1. For both Part A and Part B compare the performance of these two different algorithms
   a. Check for different n where the actual matrices are filled with random integers.
   b. Plot and find the time complexity in terms of asymptotic notation by varying input size and noting down the time required to solve this problem.
   c. Find a function *g(n)* and the associated constants for which the plot is bounded above and by *g(n)* for each of the algorithms.
   d. Find the **space complexity** for each algorithm and compare.
2. Can you identify the worst case situations for each scenario? Write your observations and derive possible conclusions about the trade-off between space and time complexity.
3. **Consider the special case of a $n \times n$ matrix A where the cells have values only from $\mathbb{Z}_2$ and the multiplication also over $\mathbb{Z}_2$. Can you find an efficient way to calculate A^n for this matrix?**