Data Structures and Algorithms
Tutorial 1

1. Write a function fibonacci_iterative(n) that generates the first n Fibonacci numbers using iteration.

2. Write a function fibonacci_recursive(n) that returns the nth Fibonacci number using recursion.

3. Implement the function pow(x, n), which calculates x raised to the power n (i.e., x^n) using binary exponentiation.

4. Implement a recursive algorithm using the **merge sort** technique to sort an array of integers in non-decreasing order.

5. Given a matrix A of size N×M, determine the maximum sum of elements that can be obtained by following a valid path from the top-left corner A[1][1] to the bottom-right corner A[N][M].
   At each step, you can move in the following directions:
   1. Down: From A[i][j] to A[i+1][j] (if i<N).
   2. Right: From A[i][j] to A[i][j+1] (if j<M).

   Your task is to compute the maximum sum of the elements encountered along any valid path.