INF1008: Data Structures and Algorithms
Tutorial 4: Recursion, DP and Greedy Algorithms

Q1. Write a recursive function $GCD(n, m)$ that returns the greatest common divisor of two integer n and m according to the following definition (recurrence relation):

$$GCD(n, m) = \begin{cases} m, if\ m \le n\ and\ n\ mod\ m = 0 \\ GCD(m, n), if\ n < m \\ GCD(m, n\ mod\ m), otherwise \end{cases}$$

Q2. Check recursively if the following objects are palindromes:
1. a word
2. a sentence (ignoring blanks, lower case and upper case differences, and punctuation marks, so that "Madam, I'm Adam" is accepted as a palindrome)

Q3. Write an algorithm to check whether a queen can be put on a certain position on a NxN chessboard. Given the function header:

def isValid(c, r, chessboard):

The function should return True if a queen can be put on column r, row r of the chessboard. Assume that chessboard is a 2D array of 0s and 1s, chessboard[i][j] == 1 means that there is a queen at row i, column j. Also assume that a queen has been placed on each row from 0 to (r-1).

Q4. You are given two eggs, and access to a 100-storey building. Both eggs are identical. The aim is to find out the highest floor from which an egg will not break when dropped out of a window from that floor. If an egg is dropped and does not break, it is undamaged and can be dropped again. However, once an egg is broken, that's it for that egg.

If an egg breaks when dropped from floor *n*, then it would also be broken from any floor above that. If an egg survives a fall from floor *n*, then it will also survive any fall lower than that.

Write (a) a recursive algorithm; and (b) a dynamic programming algorithm to compute the minimum number egg drops it takes to find the solution.

Q5. Show, in the style of the quicksort trace, how quicksort sorts the array

E A S Y Q U E S T I O N.

Note that the first element is chosen as the pivot. In the trace, only show the output of partition. You do not need to show the trace for the partition process.