Homework 3 (計算方法設計，Design and Analysis of Algorithms)

註: 請在截止日期以前透過 eeclass 線上繳交作業，請注意不接受遲交。Please submit your assignment online through eeclass before the due date. Note that late submissions will not be accepted.

Due date: April 30, 2025

1. (15%) Please find the asymptotic upper bounds of the following recurrences in big-O notation and also justify your answers. Note that you can use the master theorem to justify your answers.
   a. (5%) $T(n) = 2T\left(\frac{n}{4}\right) + 1$
   b. (5%) $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$
   c. (5%) $T(n) = 2T\left(\frac{n}{4}\right) + n^2$

2. (30%) The maximum subarray sum problem is to find a contiguous subarray with the largest sum within a given a one-dimensional array $A = [a_1, a_2, \dots, a_n]$ of n numbers. For example, if $A = [1, -2, 5, -3, 4, 8, -7, 6]$, then the contiguous subarray with the largest sum is $[5, -3, 4, 8]$ and its sum is 14. Please design a divide and conquer algorithm, whose time complexity is better than $O(n^2)$, to solve the maximum subarray sum problem (20%) and analyze its time complexity (10%). Note that you can utilize the master theorem to analyze the time complexity.

3. (30%) Given a sequence $S = [a_1, a_2, \dots, a_n]$ of n different numbers, a pair of two numbers (a_i, a_j) forms an inversion if $i < j$ and $a_i > a_j$. For example, if $S = [2,1,4,3,6,5]$, then $(2,1), (4,3)$ and $(6,5)$ are three inversions in S. The inversion counting problem is to compute the total number of inversions in S. Please design a divide and conquer algorithm whose time complexity is better than $O(n^2)$ to solve the inversion counting problem (20%) and analyze its time complexity (10%). Note that you can utilize the master theorem to analyze the time complexity.

4. (25%) Strassen's algorithm is well known to be a divide and conquer method for the matrix multiplication and its details can be found via a Google search. Please use the Strassen's algorithm to compute the following matrix product (15%). Note that you need to show not only the final result, but also the process.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}$$

   Also explain why the time complexity of the Strassen's algorithm for the multiplication of two $n \times n$ matrices is $O(n^{\log_2 7})$ (10%).