Homework 1 (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: March 26, 2025

1. (20%) Let $f(n)$, $g(n)$ and $h(n)$ be functions that are positive when n is sufficiently large. Prove each of the following statements.

(a) (10%) $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

(b) (10%) If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.

2. (30%) Binary search is a famous algorithm that can efficiently solve the problem of determining whether a given number X is in a list $a_1, a_2, \ldots, a_n$ of n sorted numbers, where $a_1 \leq a_2 \leq \cdots \leq a_n$. Show that the best-case time complexity of the binary search is $O(1)$ (5%), its worst-case time complexity is $O(\log_2 n)$ (5%) and its average-case time complexity is $O(\log_2 n)$ (20%). For simplicity, you may assume that $n = 2^k - 1$, where k is a positvie integer. Note that you can refer to page 24 of the textbook for the algorithm of the binary search.

3. (30%) Use the binary decision tree to show that finding the second largest number from a given list of n numbers needs at least $n - 2 + \lceil \log_2 n \rceil$ comparisons.

4. (20%) Consider a sequence of nine numbers 1,3,5,7,9,8,6,4,2. Demostrate how to sort these nine numbers in increasing order using the heap sort algorithm. You need to show how to construct the initial heap (10%) and also show the heaps in the process of your heap sort (10%).