1. 對於以下問題敘述，如果正確，請證明之，否則請給出一組反例。如果你選擇給出反例，僅需給出會造成矛盾的複雜度即可，不需要給出實際滿足的問題例子。
P, Q, f, g 符號代表的意思請參照作業第一頁所敘述的情境。
注意：以下所指的時間複雜度（如 $f_1(n), f_2(n), f_3(n)$）都是以 n 為該演算法的輸入大小的函數（非原始問題 P 的輸入大小）。

   (a) 如果 f 函數與 g 函數的複雜度分別為 $O(f_1(n))$ 與 $O(f_2(n))$，且 Q 問題存在 $O(f_3(n))$ 的算法，則 P 問題存在一個複雜度為 $O(\max(f_1(n), f_2(n), f_3(n)))$ 的算法。

   > 解答
   >
   > 錯誤。如果 f 函數的時間複雜度為 $O(n^2)$ ，並產生大小為 n^2 的輸出 (同時為 Q 問題的輸入)，且 Q 問題的時間複雜度為 $O(n^3)$ ，則 f 和 Q 合併的時間複雜度就已經是 $O((n^2)^3) = O(n^6)$ 了。

   (b) 如果 f 函數與 g 函數的複雜度皆為多項式，且 Q 問題存在多項式複雜度的算法，則 P 問題存在一個多項式複雜度的算法。

   > 解答
   >
   > 正確。對於一個複雜度為 $h(n)$ 的算法，可以找到一個常數 k，使得其輸出至多為 $k \cdot h(n)$，因此經過函數 f, q, g 複雜度至多為 $O(f_2(k_2 f_3(k_1 f_1(n))))$，這也是多項式。

   (c) 如果 f 函數與 g 函數的複雜度皆為多項式，且 P 問題存在多項式複雜度的算法，則 Q 問題存在一個多項式複雜度的算法。

   > 解答
   >
   > 錯誤。Q 的算法的複雜度可能至少是指數等級，例如說 $\Omega(2^n)$，這時我們會得到 P 為一個歸約至 Q 的演算法，但如此並不會造成任何矛盾，因為也許 P 問題也可以在多項式時間內歸約到另一個問題 R ，且 R 存在多項式時間的算法，此時 P 問題就存在多項式時間的算法了。

2. 請利用歸約法證明，不可能存在插入、刪除、查詢極值皆為 $O(1)$ 複雜度的類 heap 資料結構。

   > 解答
   >
   > 我們可以將排序問題規約到該資料結構的操作。
   >
   > (1) 將欲排序的序列元素全部丟進該資料結構
   >
   > (2) 不斷的取極值並刪除極值，直到該資料結構變成空的

這其實就是 heap sort 的過程。假設存在三種操作皆為 $O(1)$ 的資料結構，我們就可以在 $O(n)$ 時間進行排序，與排序問題的下界為 $\Omega(n \log n)$ 矛盾。因此，不存在這種資料結構。

3. 請將元素唯一性問題 (輸入 n 個數值，輸出 “Yes” 或 “No”，表示是否所有數值皆不重複) 線性歸約到二維的最近點對問題 (輸入二維平面上的 n 個點，輸出任意兩點之間的最短距離)。

解答

假設輸入為一個序列 $[a_1, a_2, \cdots, a_n]$，則我們可以用以下方法將唯一性問題線性規約到最近點對問題：

(1) 對每個值 a_i 構造對應的點 $(a_i, 0)$，形成點集 S。此步驟為 $O(n)$。

(2) 對點集 S 求最近點對，得到最近兩點的距離 d。

(3) 如果 $d = 0$，表示有重複的數值，此時回傳 “Yes”，否則回傳 “No”。此步驟為 $O(1)$。

4. 已知一個長度為 n 的正整數序列 $[a_1, a_2, \cdots, a_n]$，並且保證 $1 \le a_i \le n-1$。由鴿籠原理，序列中一定存在兩個位置 $i, j (i \ne j)$ 滿足 $a_i = a_j$。請找出 a_i 的值，如果有多組解，輸出任意一個解就可以。請提出一個演算法解決這個問題，並計算該演算法的時間複雜度和空間複雜度。

解答

將序列中 n 個位置視為 n 個節點，a_i 表示第 i 個節點有一個指標指向第 a_i 個節點，則這些節點將會形成許多可能帶有尾巴的環，而輸入序列中重複的值就是環和尾巴交界處的節點編號。由之前的手寫作業，我們知道只要起點不在環內，我們可以在 $O(n)$ 的時間內，使用 $O(1)$ 額外記憶體，得到尾巴環的大小，而且不會更改到序列的值。假設得到環的大小是 x，接著就從起點先沿著指標走 x 步，然後讓這個走了 x 步的指標跟另一個從起點開始的指標一起一步一步往前走，當兩者到達同一位置時，該位置就是環和尾巴的交界處。最後再觀察到，第 n 個節點一定不在環裡面 (因為 $a_i \le n-1$，所以不會有人指到它)，於是只要從第 n 個節點出發，使用尋找環的演算法就可以解決這個問題。