1 歸約法

本次作業要介紹一個用於確定兩問題可解性的相對關係的方法：歸約法。歸約法的概念大致如下：假設有兩問題 P, Q ，如果我們可以把任意一組 P 問題的輸入 w 都轉換成 Q 問題的輸入 $f(w)$ ，且 Q 問題對 $f(w)$ 的解又能夠轉換回 P 問題對 w 的解，則若 Q 問題有解，P 問題也可以透過把問題「丟」給 Q 問題解而得解。

有了概念之後，我們可以透過下列形式步驟將 P 問題「歸約」到 Q 問題：

(1) 對於 P 問題的輸入 w，構造可運算函數 f 使得 $f(w)$ 是 Q 問題的一組合法輸入。

(2) 若 Q 對輸入 $f(w)$ 有解 $q(f(w))$ ，則構造可運算函數 g 使得 $g(q(f(w))$ 為 P 對 w 的解。

其中可運算函數意指該轉換過程必須是一個算法，不能永不停止或者包含計算機不可執行的操作。我們用 $P \leq_m Q$ 表示「P 可以歸約到 Q」。

舉個簡單的例子：假設一個人 P 正在寫一份作業 w，而且他認識一位外國朋友 Q，則他可以先將作業 w 翻譯成那位朋友 Q 看得懂的語言 $f(w)$，請朋友做完作業 $q(f(w))$，最後再將答案翻譯回中文 $g(q(f(w)))$，如此一來作業就完成了！

若是進一步考慮歸約過程中的複雜度，歸約法還可以幫助我們確定複雜度的優劣性。如果 f 函數與 g 函數的運算複雜度皆為 $O(n)$，由於 $O(n)$ 為複雜度的底限，我們便可以確定 P 問題一定不比 Q 問題困難，而此時的歸約又可稱之為「線性歸約」。如果 Q 有複雜度為 $O(q(n))$ 的解，那麼 P 也至少存在一個 $O(q(n))$ 的解；反過來說，如果我們確定 P 有複雜度下界 $\Omega(p(n))$，則 Q 的複雜度也一定不低於 $\Omega(p(n))$。

以下介紹兩個歸約法應用。

例題 1

給定兩個 1 到 N 的排列 (permutation) A, B (ex: $N = 5,\ A = [1, 5, 3, 2, 4],\ B = [5, 3, 1, 4, 2]$)，求兩個排列的 LCS(Longest Common Subsequence) 長度。

解答

我們可以將原問題歸約到 LIS 問題。

假設 A 中的數字依序為 $a_1, \cdots, a_N$ ，定義函數 $f(x)$ 使得 $f(a_i) = i$。考慮序列 $B' = [f(b_1), \cdots, f(b_N)]$，則 A 和 B 的 LCS 長度就是 B' 的 LIS 長度。

例題 2

請證明凸包問題的時間複雜度為 $\Omega(n \log n)$。

凸包問題內容如下：平面上有 n 個點，求一個面積最小的凸多邊形滿足所有點

都在該凸多邊形內，並依順時針/逆時針輸出此多邊形的頂點。

解答

已知排序問題的時間複雜度下界是 $\Omega(n \log n)$，如果我們令 P 為排序問題，Q 為凸包問題，並證明 P 可以線性規約到 Q ，就可以證明凸包問題的下界為 $\Omega(n \log n)$。

對於排序問題的輸入 $w = [a_1, a_2, \cdots, a_n]$，我們可以使用凸包問題將這個序列排序：

(1) 對每個值 a_i 構造對應的點 (a_i, a_i^2)，形成點集 $S = f(w)$。此步驟為 $O(n)$。

(2) 對點集 S 求凸包。因為點的形式都是 (x, x^2)，所以每個點一定都在凸包上。

(3) 找到 $\min\{a_i\} = a_m$，並從點 (a_m, a_m^2) 開始，逆時針輸出多邊形的頂點，形成序列 $q(f(w))$。

(4) 依序將輸出的頂點序列中的每一項 (a_i, a_i^2) 對應回 a_i，形成一個序列，該序列就是原始輸入排序之後的結果 $g(q(f(w)))$。此步驟為 $O(n)$。

由以上方法我們可以將排序問題線性規約到凸包問題，於是證明了凸包問題的時間複雜度為 $\Omega(n \log n)$。

習題

1. (15 pts) 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，那麼我們可以說 P 問題存在一個複雜度為 $O(\max(f_1(n), f_2(n), f_3(n)))$ 的算法嗎？如果可以，請試著證明，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。

2. (15 pts) 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，那麼我們可以說 P 問題存在一個多項式複雜度的算法嗎？如果可以，請試著證明，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。

3. (15 pts) 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，那麼我們可以說 Q 問題存在一個多項式複雜度的算法嗎？如果可以，請試著證明，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。

4. (15 pts) 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。

5. (15 pts) 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)。

6. (30 pts) 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 $i, j (i \neq j)$ 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。

   本題評分方式：

   (a) 時間複雜度：
      (1) $O(n^2)$ 或更高：2 pts
      (2) $O(n \log n)$：5 pts
      (3) $O(n)$：10 pts

   (b) 額外空間複雜度 (不包含輸入序列)：
      (1) $O(n)$ 或更高：5 pts
      (2) $O(1)$：10 pts

   (c) 是否會更改到輸入陣列 (以程式的觀點就是該陣列是否可以宣告為 const)
      (1) 會：5 pts
      (2) 不會：10 pts