

程式語言 HW2 報告

陳俊宇 F64051114

● 執行環境步驟

此次作業我是在 ubuntu 16.04 環境中執行，作業有三個小題，我分別以 Script File 形式做成 3 個檔案，分別是 p1.pl、p2.pl、p3.pl，執行時僅需開啟終端機並輸入 `$ swipl -q -s p1.pl` 即可執行 Problem 1 的程式，以此類推。而 input 與 output 的需求跟 moodle 所附之作業說明檔幾乎相同，唯一不同處在於第一小題的 Input 輸入完之後，需要在打個 dot，如 "Input: 100."，否則無法跳出 Output 結果。

● 程式碼說明

- Problem 1:我是使用找質數的方式來實作。is_prime(N)用來檢驗 N 是否為質數。next_prime(P, P1)用來找大於 P 的整數中，最小的質數為 P1，next_prime(P, P1)在 goldbach()中是用來 query P1 value，而 goldbach()就是從 3 開始找起，若 is_prime(Input-3)亦為真，則輸出答案，否則繼續查找比 3 還大的最小質數(next_prime(3, P1))，以此類推，直到找到之質數大於等於 Input 的一半時，遞迴終止。輸出結果會印出所有可能的質數相加結果。
- Problem 2:透過 setRelation 將所有 node 的 parent 關係一筆一筆放進 dynamic database(使用 assert())，接著使用 commonAncestor(A,B,R)將 A 與 B 的共同祖先放進 R，再把 R 值放入 list 中排序，直到遞迴終止條件發生後，再將 list element 一一印出。commonAncestor(A,B,R)的實作方法為查找 A 與 B 的 parent node 是否相同，若否，則繼續遞迴呼叫 commonAncestor(A->parent,B->parent,R)；是的話，則直接輸出 R 值。但還有兩種特別狀況需考慮，A 就是 B 的 parent 或 B 就是 A 的 parent，若成立，則直接輸出 A 或 B。
- Problem 3:此題的程式架構與第二題類似，一樣將連結關係一筆一筆放進 dynamic database，而我是將雙向關係都放入資料庫(因為是無向圖)，接著透過走訪來得知是否能從起始點走到終點，而 walk function 內有一參數 V(list)，是將已走過的 node 存進去，以防止 cycle graph 造成無限走訪的情況。需特別注意的是，如果是詢問相同的兩個 nodes 是否有相連，我會直接判斷為 Yes。