

一、規格要求，違反者以零分計！

- (1) 以 Dev-C++ 或 Code::Blocks 編譯與成功執行的 C/C++ 程式碼，必須有充分的註解。
- (2) 任何一部分程式碼都不得被偵測為抄襲，程式碼的第一列要註解學號和姓名。
- (3) 檔名限以「**DS2ex5_組別_學號_學號**」開頭，**兩人一組只限繳交一份**。

二、作業內容

整合下列任務於同一個程式及單一操作選單下，未整合、無法連續執行或沒有輸入防呆措施，都各扣 5 分。若導致任務無法正常運作，該任務以零分計。

（任務零）依照有效門檻建立相鄰串列 adjacency lists

資料檔：二進位格式讀寫的檔名如 `pairs501.bin`，無標題列，每筆資料表示一個朋友關係，也代表圖的一個無向邊，包括兩個點 ID 和邊權重的 3 個欄位格式如下：

- 【學號 ID1】發訊學生的學號以 10 個字元的陣列表示。
- 【學號 ID2】收訊學生的學號以 10 個字元的陣列表示
- 【距離權重 weight】朋友的疏遠程度以浮點數 float 儲存，介於(0,1]之間的正實數，數值越接近 1 越疏遠，沒有邊相連的兩個點代表疏遠程度無限大。

有效門檻 = 給定介於(0,1]之間的正實數作為門檻，【距離權重】滿足(小於或等於)此門檻的資料(邊)稱為有效的朋友關係，才可以放入相鄰串列內。

輸入：讀取上述二進位檔，檔名如 `pairs501.bin`，使用者輸入一個【距離權重】的有效門檻。
步驟：

- (1) 逐一將【學號 ID1】及【學號 ID2】放入相鄰串列的主陣列中，每個有效的朋友關係要存入對應的**二條串列**，串列上的所有節點必須**依照學號字串由小到大排序**。
- (2) 每條串列要有至少一個有效的朋友關係(節點)，否則就從主陣列移除，**主陣列也必須依照學號字串由小到大排序**，最後將整個相鄰串列存成一個新建的文字檔。

輸出：將相鄰串列依序寫入同檔名但加註門檻值且改以 `adj` 為延伸名的文字檔，檔名如 `pairs501_0.75.adj`。

規範：每個任務違反一項各扣 5 分

- (1) 預先不知道資料筆數，禁止使用固定筆數的靜態陣列，必須採用**動態陣列**或 vector。
- (2) 必須將相鄰串列定義成專屬的 C++ 類別，並將每項任務都定義為該類別的方法成員。

（任務一）走訪無向圖找出連通成分 connected components

輸入：(只限)讀取任務零建立的相鄰串列，檔名如 `pairs501_0.75.adj`。

步驟：

- (1) 從任一節點開始以深度優先走訪 DFS 相鄰的節點，**避免走訪重複的點**，找出朋友關係圖彼此不相連的每個連通成分，連通成分涵蓋的學號**依照字串由小到大列出**。
- (2) 每個連通成分**依照所涵蓋的學號個數由多到少排序**，將所有連通成分及涵蓋的學號存成一個新建的文字檔。

輸出：依照涵蓋的學號個數由多到少將所有連通成分寫入同檔名但改以 `cc` 為延伸名的文字檔，

檔名如 pairs501_0.75.cc。

(任務二) 求出指定學生和其他學生的最短距離 shortest paths

輸入：(只限)讀取任務零建立的相鄰串列，檔名如 pairs501_0.75.adj，及使用者指定的學號。

步驟：

- (1) 列出所有學號供使用者指定一個，(只限)採用 Dijkstra's 演算法找出指定學號和同一連通成分其他學生的最短距離，最短距離是指連通兩個節點所有路徑的邊權重總和最小值。
- (2) 依照最短距離由小到大排序同一連通成分的(學號、最短距離)，並存成一個新建的文字檔，允許使用者選擇不同的學號重複執行此任務。

輸出：依照最短距離由小到大將其他學生的學號及最短距離寫入同檔名但改以 ds 為延伸名的文字檔，檔名如 pairs501_0.75.ds，不同學號的輸出結果必須累積儲存於同一檔案內。

程式碼：上機練習前一天上傳原始碼至/評量區/作業/作業 #題次 上傳/，隔天可補繳。

流程圖：上機練習前先畫好初稿，加入貼文時修正，超過期限不得更新，也不接受補繳。