

競技程式設計 HW3 報告

趙珈葦 E14056431 機械系 109

一、題目及使用語言

題目：10150 - doublet

語言：C++11

二、AC 截圖

25063710	10150 Doublets	Accepted	C++11	0.310	2020-05-24 20:20:00
----------	----------------	----------	-------	-------	---------------------

三、程式碼說明

本題使用 BFS 廣度優先搜尋達成最短路徑搜尋，並於程式開頭定義 Queue 相關代碼，含 struct Queue, enqueue(), dequeue()。另外亦定義一函式 isDoublet(string, string)，比對兩字串不同處數量以辨別出 doublet pairs。

資料儲存的部分，由於考慮兩不同長度字串不形成 doublet pair，因此考量效率問題後，認為若單筆 start 和 end 為相同長度字串時，也應當搜尋同長度資料即可，因此使用一個含 16 個 Vector<string>的陣列分開儲存不同長度之字串。

接下來是程式主體，流程如下：

1. 資料讀取及儲存：

使用 while(getline(cin, str))讀取各行，由於根據題意以一空行區隔資料集及題目，因此於每次讀取後判斷讀入資料字數作為迴圈終止條件，即長度為 0(空行)

2. 讀取開頭及結尾字串：

若兩者長度不一可直接判定結果為 No solution。

3. 建圖：

以一 adjacent matrix 表示各字串間 doublet 關係，和資料集儲存概念相同，將各長度字串集分開處理及儲存，並考量效率，僅做上三角的部分以節省運算時間。

4. 進行 BFS：

由 last word 作為起點，找到透過 adjacent matrix 可到達的字串，更改其 color 及 predecessor 並 enqueue，接著透過 dequeue 依序處理 queue 中的所有字串。

5. 結束 BFS：

若 queue 變為空(無法到達)，則輸出 No solution。

若在 enqueue 時偵測到已是 first word，則在改寫完 predecessor 後脫離迴圈，並根據各字串的 predecessor 依序印出路徑。

6. 重複解題至結束：

Go to 2.，直到輸入終止。