

Programming Assignment 4

Implement the Bellman-Ford algorithm

2019.05.14

《題目》

給定一有向圖 $G(V, E)$ ，參考 Slide unit 9 P.8 實作 Bellman-Ford shortest path 演算法，並利用 Slide unit 9 P.12 修改原本的演算法，加快 Bellman-Ford 的執行時間，最後依照要求輸出答案。

《Input 說明》

每筆測資有以下資訊：

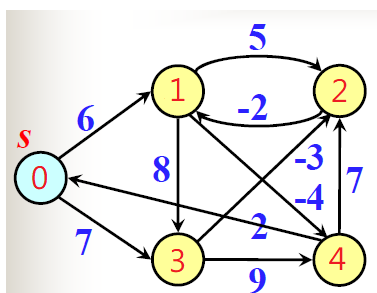
1. 第一行給定圖 G 的總點數 V 與總邊數 E
2. 第二行給定 Source 與 Destination
3. 剩下 E 行提供有向邊的資訊（起點、終點、權重）

《Output 說明》

每筆測資有以下資訊：

1. 第一行輸出原版 Bellman-Ford 的執行時間（以 ms 為單位）。
2. 第二行輸出你實作的加速版 Bellman-Ford 的執行時間（以 ms 為單位）。
（若沒實作加速版則不需要輸出此行）
3. 若圖沒有 Negative weight cycle，依序輸出 Input 給的 source 與 destination，這兩者之間的最短路徑上經過的所有點，並在下一行輸出其路徑長。
4. 若圖有 Negative weight cycle，先輸出「There is a negative weight cycle.」，接著在下一行依序輸出此 cycle 的所有點。
（注意：圖中只會有唯一一個 Negative weight cycle，輸出 Negative weight cycle 時沒有限制起點，但需依照邊的方向輸出。）

《範例測資》



註：第一筆範例測資為 Slide unit 9 P.10-11（左圖）

輸入	輸出 (時間為範例)
5 10 0 4 0 1 6 0 3 7 1 2 5 1 3 8 1 4 -4 2 1 -2 3 2 -3 3 4 9 4 0 2 4 2 7	original time: 3000 ms modified time: 1500 ms 0 3 2 1 4 -2
4 5 0 3 0 1 1 0 2 2 1 2 3 2 3 -4 3 1 -1	original time: 1500 ms modified time: 800 ms There is a negative weight cycle. 1 2 3

《報告內容》

報告需包含：

- 改進 Bellman-Ford 演算法的哪些部分（例如：Relax 時不用看所有的邊），請使用 Pseudo code 的方式寫出來（若你有實作），並在程式碼中註解實作加速的部分
- 原版與加速版的時間比較
- 輸出 Negative weight cycle 的方式
- 分析加速版的演算法複雜度

並請將你輸出的答案以最清楚明瞭的方式呈現。

《作業繳交及注意事項》

1. 可使用的程式語言為 C、C++、Java；需繳交你的程式之原始碼，原始碼中重要演算法部分需有註解，若使用專案則為整個專案檔。

2. 繳交一份報告，報告需包含的內容如上，Word 或 PDF 皆可。
3. 如果你有其他可以說明你的作法的檔案（影片...等），也可以一起繳交上傳。
4. 將所有繳交上傳的檔案，包成一個壓縮檔上傳至 LMS 作業區；檔名為「PA04_學號_姓名.(壓縮檔副檔名)」(ex. PA04_106502666_王小明.zip)，壓縮檔內至少要有 2 個檔案：程式碼 及 此次作業的報告說明。
5. 繳交期限：**06/02(日) 23:55**

《評分標準》

程式碼 80%（原版 50%、加速版 30%）、報告 20%（原版 10%、加速版 10%）

編譯不過、未實作出演算法：0 分

上傳檔名不符格式：扣 5 分

如有抄襲一律 0 分計算（包含抄襲網路）