國立臺南大學資訊工程學系

資工三「演算法」課程

第四次作業

**題目: 移動軌跡相似問題**

|  |  |  |
| --- | --- | --- |
| 班級 | ： | 資工三 |
| 姓名 | ： | 林晉昇 |
| 學號 | ： | S10359004 |

老師：陳宗禧

中華民國 105年12月26日

# 目錄

1. **簡介及問題描述……….……………..…………………………………………3**
   1. **簡介…………….…………………………………………………………………………3**
   2. **問題………………………….……………………………………………………………3**
2. **理論分析與演算法則…………..………………………………………4**
3. **程式設計環境架構.………………………..…………………………………6**
4. **程式.…………………………………………..………………………………7**
5. **執行結果、討論與心得.………………………..……………………………9**

參考文獻………………………………………………………….…………………11

**(一) 簡介及問題描述**

設計與實作計算Longest Common Sequence，理論驗證與實驗分析該問題!

1. 簡介

實作動態規劃演算法來找到最長的共同子序列。

2. 問題

(1) 同學一天中騎機車(or 開車)到過的地點與駐留時間，形成一按照時間順序的序列。以動 態規劃技巧實作 LCS (Longest Common Subsequence) 演算法，利用該程式分析兩位同 學一天移動軌跡的相似度，並分析其執行效率。假設兩個位置一樣，但時間不一樣， 其值定義為 min{𝑡1,𝑡2} max{𝑡1,𝑡2} 。如： (i) 同學 u1 一天的移動軌跡 L1 (10 個地方)： <南大宿舍, 30 分鐘>, <早餐店, 30 分鐘>, <南大榮譽校區, 3 小時>, <ㄚ禧仔, 40 分鐘>, <南大宿舍, 兩小時>, <南大操場, 60 分鐘>, <南大餐廳, 60 分鐘>, <大東夜市, 2 小時>, <安平觀夕平台, 120 分 鐘>, <南大宿舍, 60 分鐘>

(ii) 同學 u2 一天的移動軌跡 L2 (8 個地方)： <南大宿舍, 50 分鐘>, <早餐店, 60 分鐘>, <南大府城校區, 2 小時>, <南大餐廳, 60 分鐘>, <新光三 越, 兩小時>, <安平老街, 60 分鐘>, <安平觀夕平台, 90 分鐘>, <南大宿舍, 40 分鐘>

共同子序列的值 LCS-length 計算如下： 30 50 + 30 60 + 60 60 + 90 120

+ 40 60 = 3.52 南大宿舍 早餐店 南大餐廳 安平觀夕平台 南大宿舍 相似度定義為 S(u1, u2)= 𝐿𝐶𝑆−𝑙𝑒𝑛𝑔𝑡ℎ max⁡ {𝑙𝑒𝑛𝑔𝑡ℎ(𝐿1),𝑙𝑒𝑛𝑔𝑡ℎ(𝐿2)} ，值介於 0~1。S(u1, u2)值越大，表示兩位 同學移動軌跡 L1 與 L2 的相似度愈高，反之，相似度愈低。上例的相似度計算方式為 S(u1, u2)= 3.52 max⁡ {10,8} =0.352。最後，列出兩位同學最長共同到過的地方(最長共同子序列)。

(2) 全班有 k 位同學，設計一程式找出哪兩位同學一天的移動軌跡之相似度最高，並算出 其值，分析其執行效率。 (3) 全班有 k 位同學，設計一程式找出哪些同學一天的移動軌跡之相似度與你最相近，但 相似度要大於或等於一門檻值δ, 0 ≤ δ ≤ 1，並分析其執行效率。

[Note: 實驗時 k 值至少要大於 50，同學移動軌跡至少大於 5 以上。] **(二) 理論分析與演算法則**

Longest Common Sequence

中文稱作「最長共同子序列」，簡稱LCS，所謂的LCS指的就是說假設給定兩個序列，由兩個序列共同的元素能夠組合而成最長的序列(相對位置不能與原本的序列不同)。

例如: s1: president

s2: providence

LCS: priden

p r e s i d e n t

↓ ↓ ↓ ↓ ↓ ↓

p r o v i d e n c e

**方法1 暴力法**

找出s1的子序列，逐一檢查是否為s2的子序列

假設s1長度m，s2長度n，則s1有種子序列(每個元素選與不選)；檢查s2是否與s1的一個子序列相同需要O(n)的時間複雜度，所以worst case時間複雜度是O(n)。

**方法二 動態規劃**

要找出s1與s2的LCS，表示為LCS(s1, s2)。

首先要分割問題，拆解s1與s2的最後一個元素:

s1 = sub1 + e1

s2 = sub2 + e2

可以分為以下四種情況:

1. LCS包含e1，不含e2 →　LCS(s1, s2) = LCS(s1, sub2)
2. LCS包含e2，不含e1 →　LCS(s1, s2) = LCS(s1, sub2)
3. LCS不包含e1及e2 →　LCS(s1, s2) = LCS(sub1, sub2)
4. LCS包含e1及e2 →　LCS(s1, s2) = LCS(sub1, sub2) + e1 (e1與e2相等)

透過以上的分析，可以得到以下遞迴公式:

LCS(s1, s2) = LCS(s1, sub2) or LCS(s1, sub2) or LCS(sub1, sub2)，當e1≠e2

LCS(s1, s2) = LCS(sub1, sub2) + e1，當e1=e2

簡化後可以得到:

LCS(s1, s2) = max(LCS(sub1, s2), LCS(s1, sub2))，當e1≠e2

LCS(s1, s2) = LCS(sub1, sub2) + e1，當e1=e2

找出了遞迴的關係後，接著要算的就是LCS的長度與列出其中一個LCS。

計算LCS長度的演算法pesudo code如下:

**Algorithm** LCS-Length(A, B)

For i←0 to m do len(i, 0) = 0

For j←1 to n do len(0, j) = 0

For i←1 to m do

for j←1 to n do

if =

then len(i, j) = len(i-1, j-1) +1

prev(i, v) = ↖

else if len(i-1, j) len(i-1, j)

then len(i, j) = len(i-1, j)

prev(i, j) = ↑

else len(I, j) = len(i, j-1)

pre(i, j)= ←

return len and prev

**動態規劃求LCS演算法的時間複雜度為O(mn)**

backtracing algorithm

**Algorithm** Output-LCS(A, prev, i, j)

if i = 0 or j = 0 then return

if prev(i, j)=↖

then Output-LCS(A, prev, i-1, j-1)

print

else if prev(i, j) = ↑ then Output-LCS(A, prev, i-1, j)

else Output-LCS(A, prev, i, j-1)**(三) 程式設計環境架構**

程式設計語言、工具、環境與電腦硬體等規格說明…

1. 程式語言

C++

2. 程式開發工具

Visual Studio C++ 2015 Community

3. 電腦硬體

處理器: Intel Core i7-4500U

RAM: 8GB

系統類型:64位元作業系統，x64處理器**(四) 程式 (含source code, input code, and output code)**

1. 主程式

Traj-Sim \_Prog\_S10359004.cpp

2. Input Code Format

第一題的答案已經顯示

只要輸入第二題的人數以及第三題的人數(2~100)，以下以36及54為例:

36

54

3. Output Code Format

輸出三題的結果

(1)隨機產生2位同學的移動軌跡與駐留時間的序列，分析兩位同學移動軌跡的相似度

結果:

LCS路徑為: 早餐店->夢時代->榮譽校區->觀夕平台->杜卡勒小廚

相似度: 0.19611

Time: 0.0013877384s

----------------------------------------------------------------------------------

(2)隨機產生n位同學的移動軌跡與駐留時間的序列，並找出哪兩位同學的移動軌跡相似度最高

輸入人數(介於2~100): 36

結果:

同學4與同學25的相似度最高

LCS路徑為: 大東夜市->早餐店->觀夕平台->杜卡勒小廚

相似度為: 0.380205

Time: 0.1582209921s

----------------------------------------------------------------------------------

(3)隨機產生n位同學的移動軌跡與駐留時間的序列，並找出哪位同學與你的的移動軌跡相似度最高

輸入人數(介於2~100): 54

我的路徑為:南大宿舍->杜卡勒小廚->大東夜市->榮譽校區->觀夕平台->黃金海岸->學餐->府城校區->南大宿舍

相似度的最低門檻值預設為: 0.3

結果:

雖然最接近的相似度小於最低門檻值，但是我都已經找了那麼久，看在我的面子上還是輸出一下移動軌跡與你最相近的同學吧!

同學42與你的的相似度最高

LCS路徑為: 南大宿舍->杜卡勒小廚->大東夜市->觀夕平台->黃金海岸->學餐->府城校區->南大宿舍

相似度為: 0.261035

Time: 0.0394288762s

**(五) 執行結果、討論與心得**

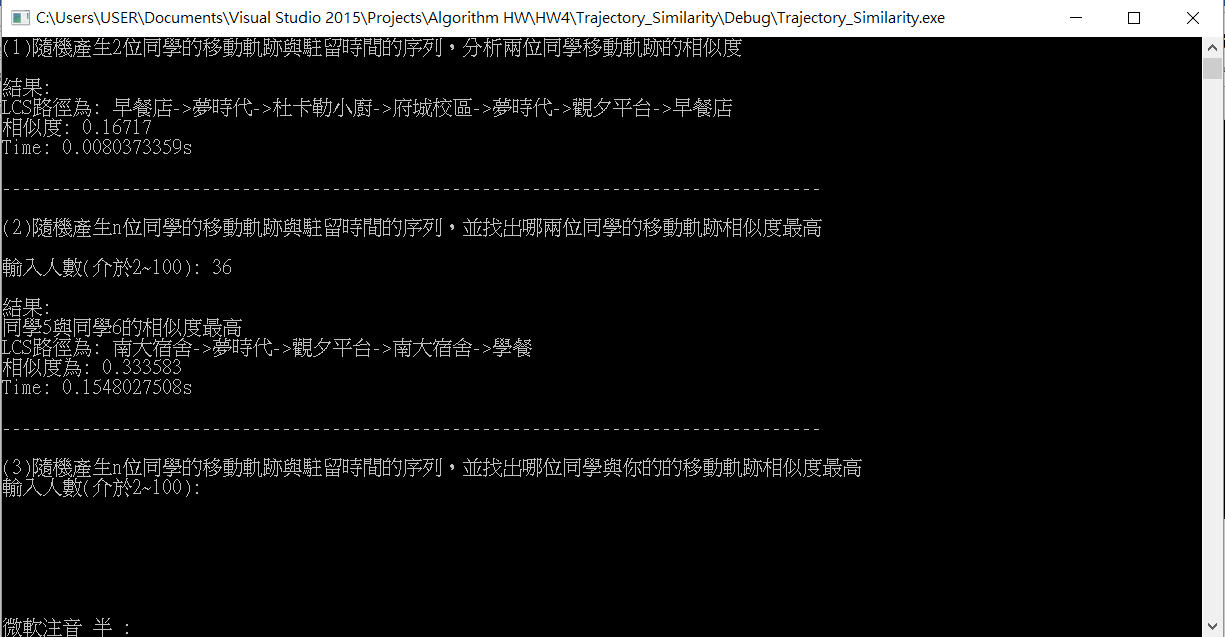
1. 執行結果

(1)程式執行時的狀況(已經算出第一題的答案)



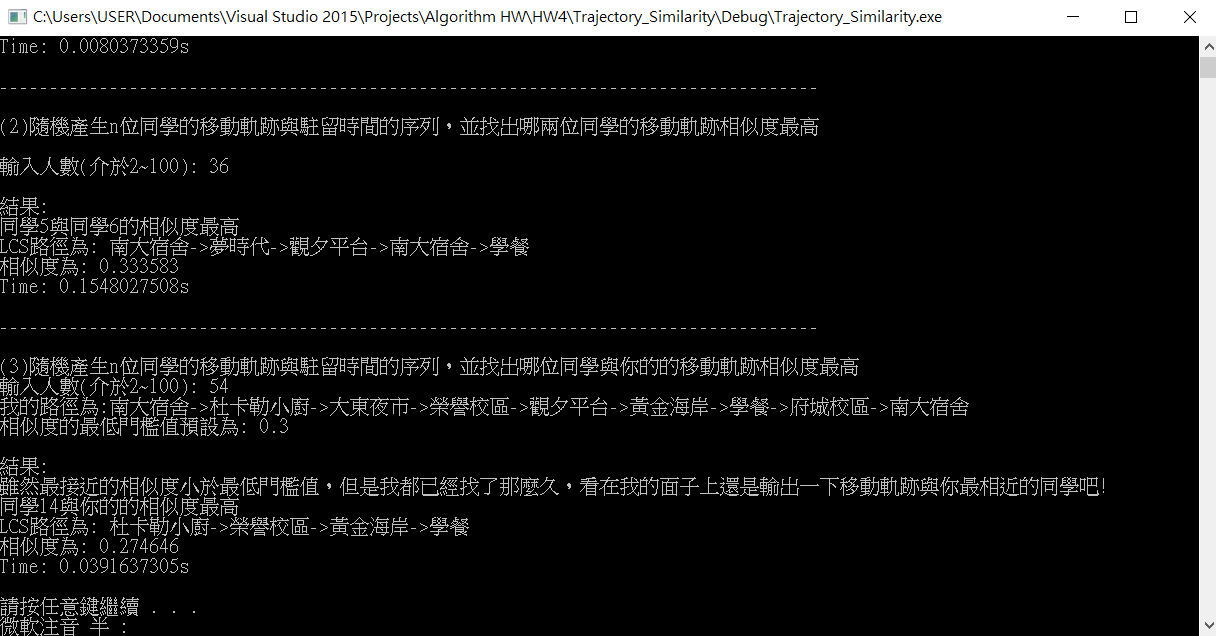
(2)輸入: 36

輸出第二題的結果



(3)輸入54

輸出第三題的結果



2. 程式說明

(1) 所有同學的路徑將根據輸入的人數隨機產生

(2) 自己的路徑以及相似度門檻值已經在程式內宣告完畢

3. 心得

經過這次作業的洗禮之後，我終於了解動態規劃的強大之處了，原本複雜的遞迴關係，竟然能夠透過動態規劃的方式實作出來。這次的作業也讓我真正理解動態規劃與回朔法的實作會遇到的問題以及撰寫的技巧，希望下次遇到這類問題時也能夠迎刃而解。**參考文獻**

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms," Second Edition, The MIT Press, 2001.
2. Anany V. Levitin, "Introduction to the Design and Analysis of Algorithms," 3nd Edition, Addison Wesley, 2012.