國立臺南大學資訊工程學系

資工三「演算法」課程

第二次作業

**題目: City Map**

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

老師：陳宗禧

中華民國 105年10月30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….………………………..………………………………………4**
3. **演算法則….…………………………..……………………………………6**
4. **程式設計環境架構.………………………..…………………………………8**
5. **程式.…………………………………………..………………………………9**
6. **執行結果、討論與心得.………………………..……………………………14**

參考文獻………………………………………………………….…………………16

**(一) 簡介及問題描述**

1. 簡介

給定n個充電站以及電動機車的座標位置，利用暴力法求出最短、最長距離、Convex Hull，以及利用匈牙利演算法解決Assignment問題。

2. 問題

給定一地圖，設有*n*個u-motor充電站(還電動機車與充電站)，例：充電站設置在台南古蹟或車站旁等。假設目前有*n*個人騎u-motor(電動機車)，想找充電站，底下有幾個問題要解決：

(1) 哪兩個充電站靠最近? 距離多少? (brute-force algorithm)

(2) 哪兩台電動機車靠最近? 距離多少? (brute-force algorithm)

(3) n個充電站的範圍有多大(Convex-Hull，它的面積以及最遠的距離)? (brute-force algorithm)

(4) n台電動機車的範圍有多大(Convex-Hull，它的面積以及最遠的距離)? (brute-force algorithm)

(5) 假設每一個充電站只有一個空位，要以最省電的方式*n*個人騎u-motor到充電站，如何配對，以及求出最少的總耗能? (Hungarian method)

**Notes:**

(1) 輸入(Input) n個二維的座標充電站與電動機車的位置。

(2) 請自行定義距離(兩點間)以及電動機車的耗能(假設電動機車每公里耗電量為*x kWh/km*，如0.024 kWh/km，盡可能列出參考文獻)。

**(二) 理論分析**

1. 充電站間最短距離

要在n個點之中求出最短之兩點的距離，可以利用窮舉法列出所有任兩點間的距離後，找出最小的值即為最短距離。

要使用兩點的座標求出兩點間的距離可以用Euclidean distance來表示，即

d(, ) =

由於兩個數開根號後，相對大小並不會改變，因此根號的部分可以捨去，使用平方相加後的值來進行比較即可。

2. 電動機車間最短距離

同第1小題。

3.充電站之 Convex Hull面積及最遠距離

Convex Hull是能夠包覆n個點的一個外殼之中，面積最小的一個，而各個外殼一定是一個凸多邊形。

而所謂「凸」多邊形的定義就是圖形內任兩點的連線不會經過圖形的外部。

而要求出Convex Hull的各個頂點我們首先也要知道一個性質:Convex Hull相鄰兩點之間的連線，會使其他點全部落在直線的某一側。也就是說只要檢查任兩點之間的連線能讓剩餘的所有點都落在直線的一側，這兩點一定是Convex Hull的頂點。

那要如何檢查他的點都落在直線的一側呢?一個方法就是求出這兩點的直線方程式，接著將剩餘的所有點一一帶入此直線方程式，若帶入的結果全部皆為同號，那麼就代表帶入的點全部都落在直線的某一側。

求出Convex Hull所有的頂點後接著要求的是它的面積，要求多邊形面積的方法有很多種，我使用的是利用行列式的解法，原理是用相鄰兩點的座標來計算差積，公式如下:

Area = \*

如此一來就可以求出Convex Hull的面積，至於要求最遠距離的話，則是利用同1、2小題的方法，算出任兩點間的Euclidean distance後，找出最大的值，即是最遠距離。

4. 電動機車之Convex Hull面積及最遠距離

同第3小題。

5. 充電站與電動機車間的Assignment Problem

對於每一個充電站，都分配一台電動機車，使得電動機車騎到充電站所產生的總耗能最小，因為每台電動機車的耗能均相同，亦即求出電動機車與充電站的總距離為最小。

除了一般的暴力法(時間複雜度O(n!))外，還可以使用匈牙利法(時間複雜度O())來解決指派的問題。

**(三) 演算法則**

1. 第一個演算法(Algorithm)

BruteForceClosestPair(P)

//Input: A list P of n (n2) points (, ), …. , (, )

//Output: The distance between the closest pair of points

d

for i1 to n – 1 do

for j i + 1 to n do

d min (d, +)

return d

1. 演算法時間複雜度(time complexity)

假設最基本的運算是c

那麼T(n) = = c = c[(n-1)+(n-2)…+1] = cn(n-1) O(n)

1. 演算法空間複雜度(space complexity)

空間複雜度為S(1)

2. 第二個演算法(Algorithm)

同第1小題。

3. 第三個演算法(Algorithm)

求Convex Hull之面積及最遠距離

//Input: A list P of n (n2) points (, ), …. , (, )

//Output: The area of the polygon and the distance between the farthest pair of points

1. for i1 to n – 1 do

for j i + 1 to n do

for k 1 to n do

將代入及所產生的直線方程式中

若所有的點都在直線的同一邊，將及這兩點的座標位置存起來

1. 找出最左下角的點，方法是先找到所有點之中的x值最小的那個點，若有超過一個點的x值相同，就再比較那些點的y值，取y值最小的即為答案。
2. 利用左下角的點當作基準點，算出左下角的點與所有剩下的點的斜率，將斜率做排序(由小到大)，就可以得到將點從逆時針排序的順序。
3. 利用行列式方法就可以求得面積Area = \* 。
4. 最後用與第1、2小題相同概念的暴力法來求得最遠的兩點。
5. 演算法時間複雜度(time complexity)

我的演算法時間複雜度為O()

1. 演算法空間複雜度(space complexity)

空間複雜度為S(1)

4. 第四個演算法(Algorithm)

同第3小題

5. 第五個演算法(Algorithm)

匈牙利演算法(Hungarian Algorithm)

1. 建立每台機車與每座充電站之間距離的表格
2. 在各列中找最小值，將該列中各元素減去此值。
3. 在各行中找最小值，將該行中各元素減去此值。
4. 檢查各列，對碰上之第一個零，做記號，同列或同欄的其他零則畫X(~~0~~)(由零較少的列先做，可不依順序)。
5. 檢查可否完成僅含零的完全指派，若不能，則畫出最少數目的垂直與水平的刪除線來包含所有的零至少一次。
6. 找出未被畫線的元素中之最小值K，將含有此些為被畫線的元素的各列所有元素減去K，若造成負值，則將該欄加上K，形成新矩陣後，回到步驟(4)。
7. 演算法時間複雜度(time complexity)

時間複雜度為O()

1. 演算法空間複雜度(space complexity)

空間複雜度為S(1)

**(四) 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 in MS Windows

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**(五) 程式 (含source code, input code, and output code)**

程式含source code, input code, and output code等…

1. 主程式

CMap \_Prog\_S10359004.cpp

2. Input Code Format

檔案輸入的格式如下:

一個正整數代表有N筆資料(N<100)

N組(X,Y)座標的值分別代表N座充電站的位置 (X,Y均為整數)

N組(X,Y)座標的值分別代表N台電動機車的位置

//第K組(X,Y)座標代表第K座充電站或第K台電動機車

Three of examples for input use are in below….

輸入1或2或3(代表讀入Map1或 Map2或 Map3)

1. Map1.txt

10

455 741

293 889

543 141

875 432

769 258

-45 -52

-47 586

200 500

250 -84

101 -50

677 964

29 658

813 810

-77 46

85 45

-52 43

77 79

540 -320

250 35

-12 -12

1. Map2.txt

4

5 7

4 3

-2 1

2 -5

10 3

6 7

9 -2

5 5

1. Map3.txt

8

70 -83

-26 -46

38 93

-88 -39

-33 -27

-47 -53

42 32

2 -64

83 -29

-33 72

16 76

-53 -76

-75 -88

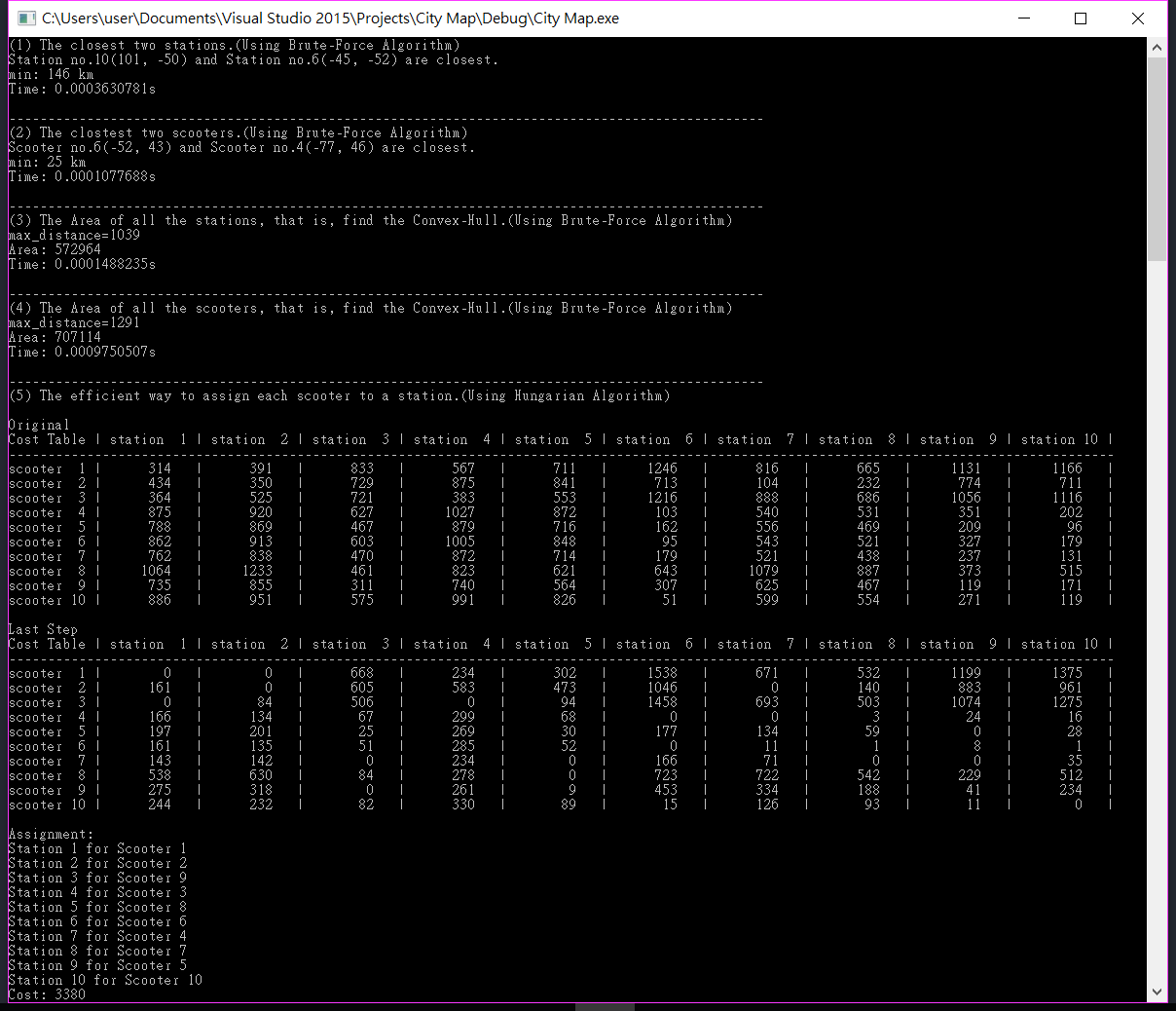
-7 -34

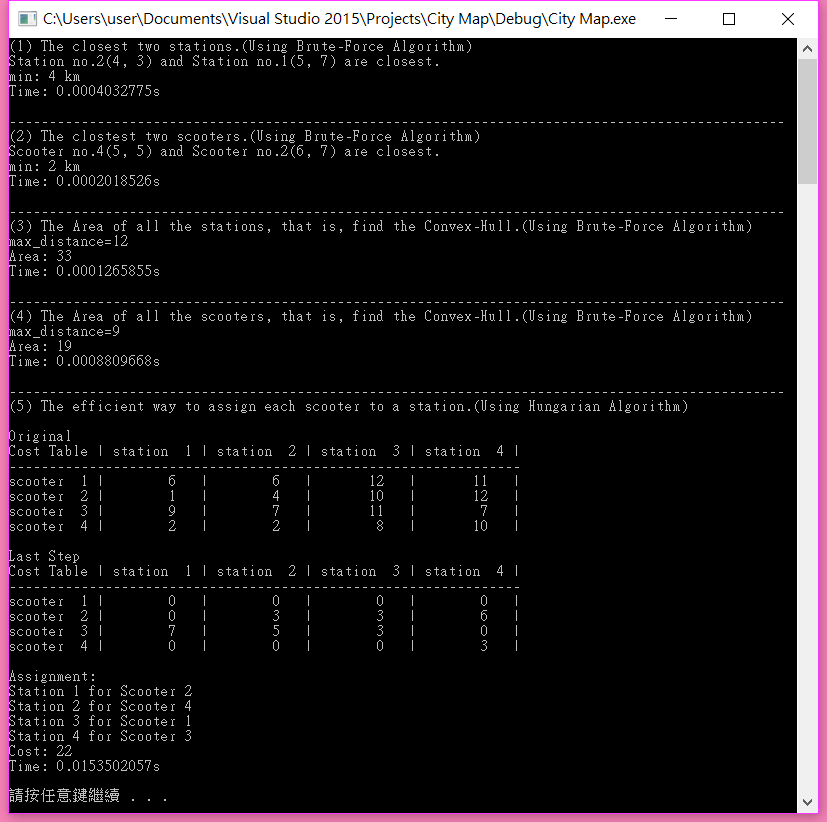
-45 -72

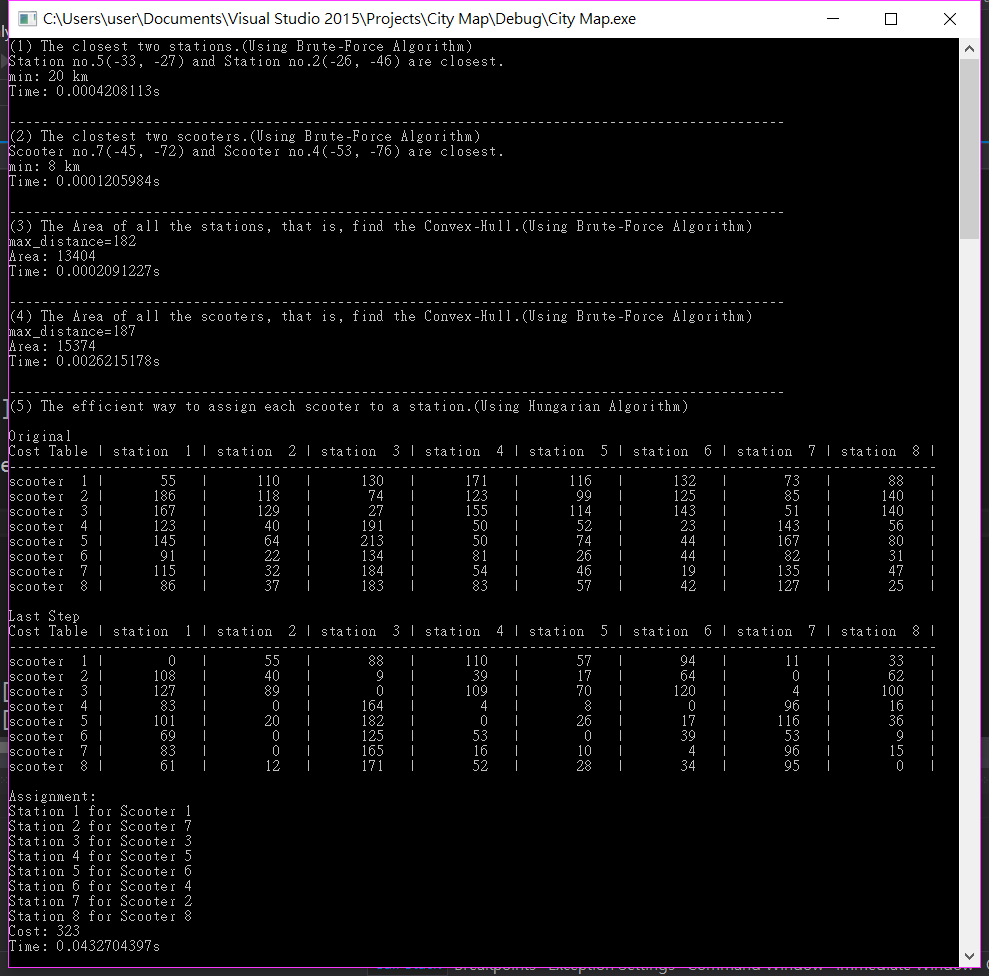
-16 -82

3. Output Code Format

Three of examples for output use are in below….

(1)

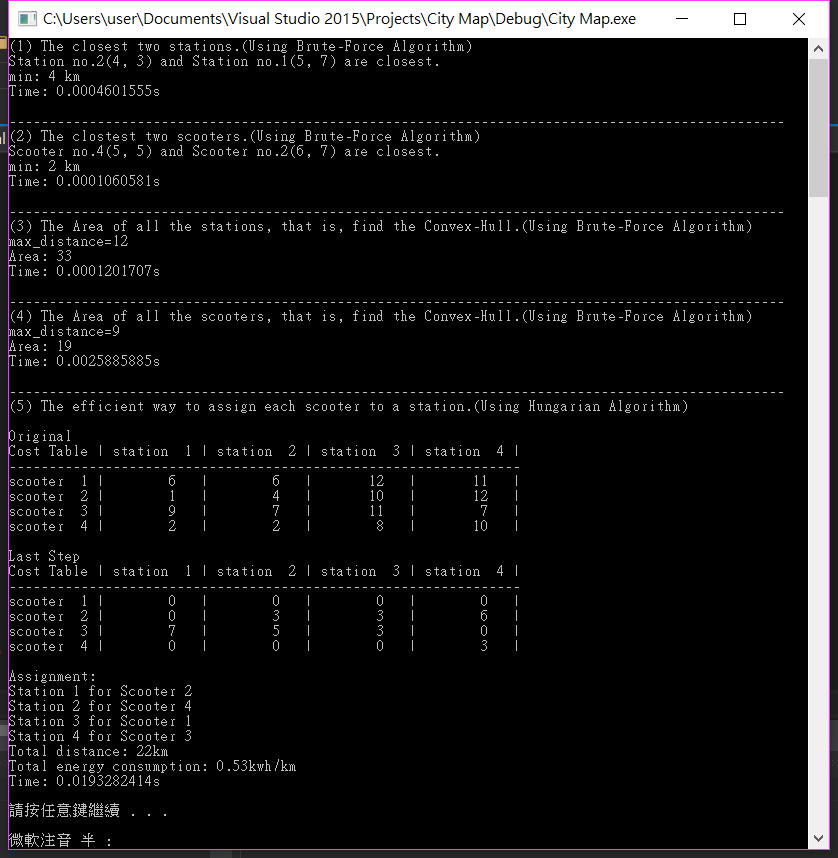
(2)

(3)

**(五) 執行結果、討論與心得**

執行結果與討論 (執行時間、problem *n*的大小等問題討論)等…

1. 執行結果



以第二個input為例，輸入為4個充電站的座標以及4台電動機車的座標。

1. 計算出最近兩點的充電站的距離為(4, 3)及(5, 7)兩點，距離是4公里。
2. 計算出最近兩點的電動機車的距離為(5, 5)及(6, 7)兩點，距離是2公里。
3. 計算出Convex Hull的面積是33，最長距離為12。
4. 計算出Convex Hull的面積是19，最長距離為9。
5. 顯示電動機車與充電站距離關係的表格，以及完成計算的表格。接著是每台充電站對應到的機車，最短里程數跟最小耗能。

2. 心得

這次的報告對在下來說真的是一個莫大的挑戰，理解演算法是一回事，要將演算法實作成程式碼又是另一回事了。簡而言之，這次的作業考驗的真的是程式撰寫的基本功了!特別是第五小題，在下簡直是花費了畢生之所學，費盡千辛萬苦才完成的!(雖然好像是錯的...)不過我覺得這沒有關係，重要的是我在撰寫程式的過程中更加了解了演算法的奧妙，特別是在下還自創了一招叫做: sorting 中的sorting，這方法用在第3題以及第5題，意思就是說在排序一個陣列的同時順便排序另一個陣列，當然這兩個陣列之中必須存在有某種特殊關係，我覺得這真的聽起來有點玄，不過一時之間我也不知道怎麼解釋，好吧，只要能夠心領神會就好了!

不過在下是不會放棄的，不論是再艱深的演算法，在下一定會披荊斬棘，如入無人之境般擊退他們的!(希望啦...)希望在下能繼續堅持走完這艱難的演算法之路!**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.
3. Chapter\_3\_Brute\_Force\_and\_Exhaustive\_Search.pdf (老師上課PPT)