



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：Travel Map

班級：資工三

姓名：呂益銓

學號：S10659013

老師：陳宗禧

中華民國 108 年 10 月 30 日



# 目錄

(一) 簡介及問題描述.....	3
1. 簡介.....	3
2. 問題.....	3
(二) 理論分析.....	4
(三) 演算法則.....	5
(四) 程式設計環境架構.....	9
(五) 程式.....	10
(六) 執行結果、討論與心得.....	16
參考文獻.....	19

# (一) 簡介及問題描述

## 1. Problem: Travel Map

給定一地圖，內有  $p$  個興趣點(POI, Point of Interest)，例：臺南古蹟、臺南小吃等。假設目前有一個人或一個旅行團，對於臺南市的美食或古蹟有高度興趣，並且要規劃行程，但對於該行程需要有資訊系統輔助，給定行前的建議，底下問題為規劃者需要知道的答案：

- (1) 哪兩個 POIs 靠最近？距離多少？(brute-force algorithm)
- (2) 這些 POIs 的範圍有多大 (Convex-Hull，它的面積以及最遠的距離)？(brute-force algorithm)
- (3)
  - (a) 假如要到所有  $p$  個 POIs，則最短行程距離是多少？  
(exhaustive search + brute-force algorithm)
  - (b) 假如我們設計一個新演算法：(Convex-Hull-TSP Algorithm)
    - i. 求出所有點的 Convex-Hull
    - ii. 除 Convex-Hull 上的點外，其餘 POIs 找出離 Convex-Hull 邊最近的點投影
    - iii. 按照投影點依序由 Extreme Point 點開始旅遊，再繞回起始點 請問該演算法的時間複雜度(以 big O 表示)以及(行程)花多少距離？
  - (c) 比較 3(a)和 3(b)兩者的距離比較，以及執行時間比較？
- (4) 假設我們不可能到全部的 POIs，需要縮短行程，則要找多個 POIs 間的距離不超過一固定範圍( $r$ )形成一個 Group (Clique)  $g$ ，假如 Group  $g$  內的 POIs 個數低於  $k$  個，也不安排到  $g$  的行程。假設符合規範的 Group 共有  $n$  個， $g_i$ ,  $1 \leq i \leq n$ ，列出這  $n$  個 Group 的中心位置  $c_i$ ,  $1 \leq i \leq n$  (或指定該群的任一點 POI or 最靠近的停車場等)。更進一步，我們想要知道假如規劃由一點  $c_j$  開始出發(Group 間開車)，要玩遍所有的 Group 內的 POIs (Group 內走路)，再回到出發處。請問該行程(開車與走路)要花多少距離？(exhaustive search + brute-force algorithms: Clique, TSP problems (TSP 方法用 3(a) or 3(b)亦可，兩者皆實作亦可))

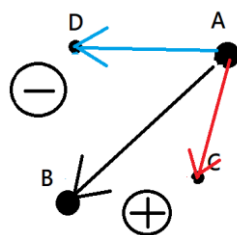
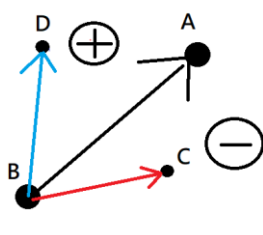
## (二) 理論分析

### 1. 哪兩個 POIs 靠最近？距離多少？

在圖上任取兩點組合，假設有  $n$  個 poi，那會有  $(n(n-1))/2$  種組合，從每種組合都算出兩點距離找出最小值

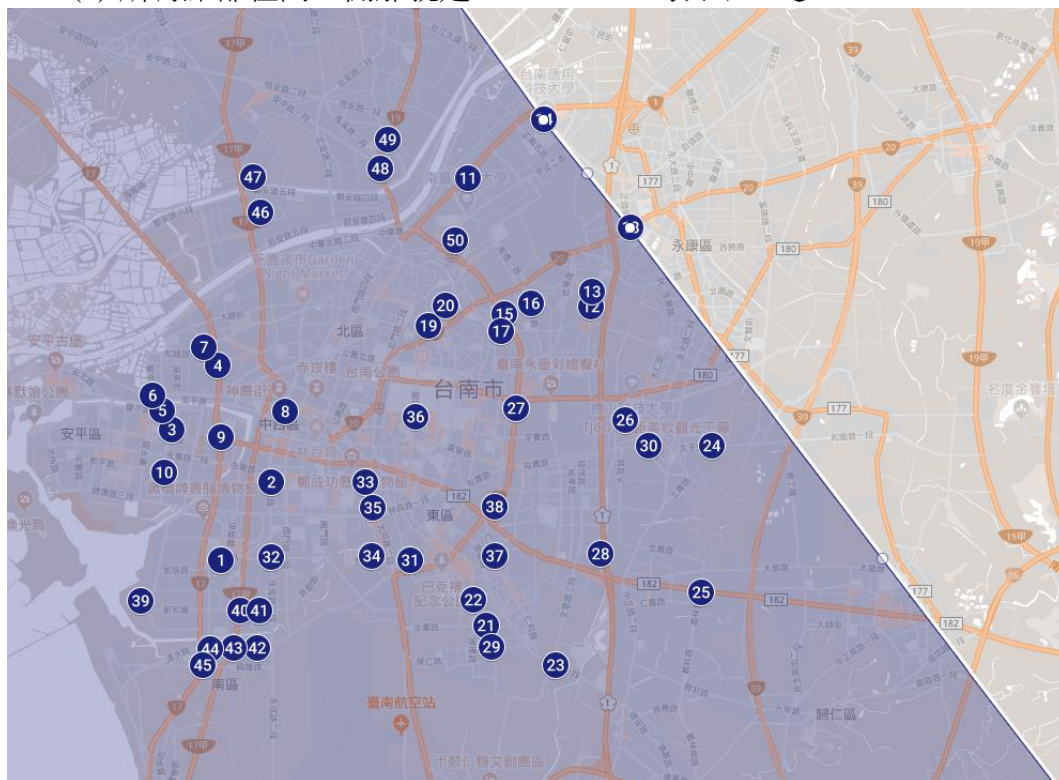
### 2. 這些 POIs 的範圍有多大 (Convex-Hull，它的面積以及最遠的距離)? (brute-force algorithm)

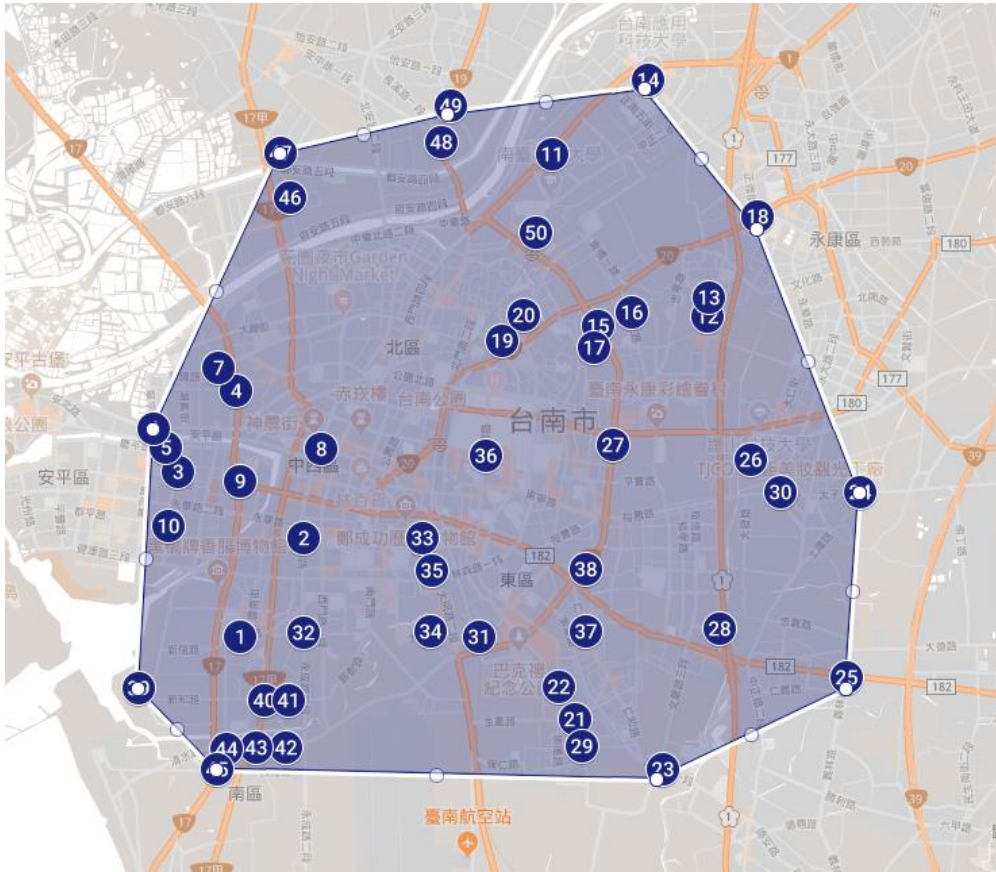
(1) convex-hull: 在圖上任取兩點組合連成一條線，假設有  $n$  個 poi，那會有  $(n(n-1))/2$  條線，每條線判斷剩餘的點是否都在線的同一側，如果都在同一側那就是 convex-hull 的其中一邊，判斷是否同一側利用外積的方法，如果兩條向量外積  $> 0$ ，那點就會在線的左半邊，如果小於 0，就會在右半邊，等於 0 就在線上，如果點都在同一側那算出來的外積都會是同個符號，藉此判斷是否點都在同一側



Example:

(1) 所有點都在同一側那就是 convex-hull 的其中一邊



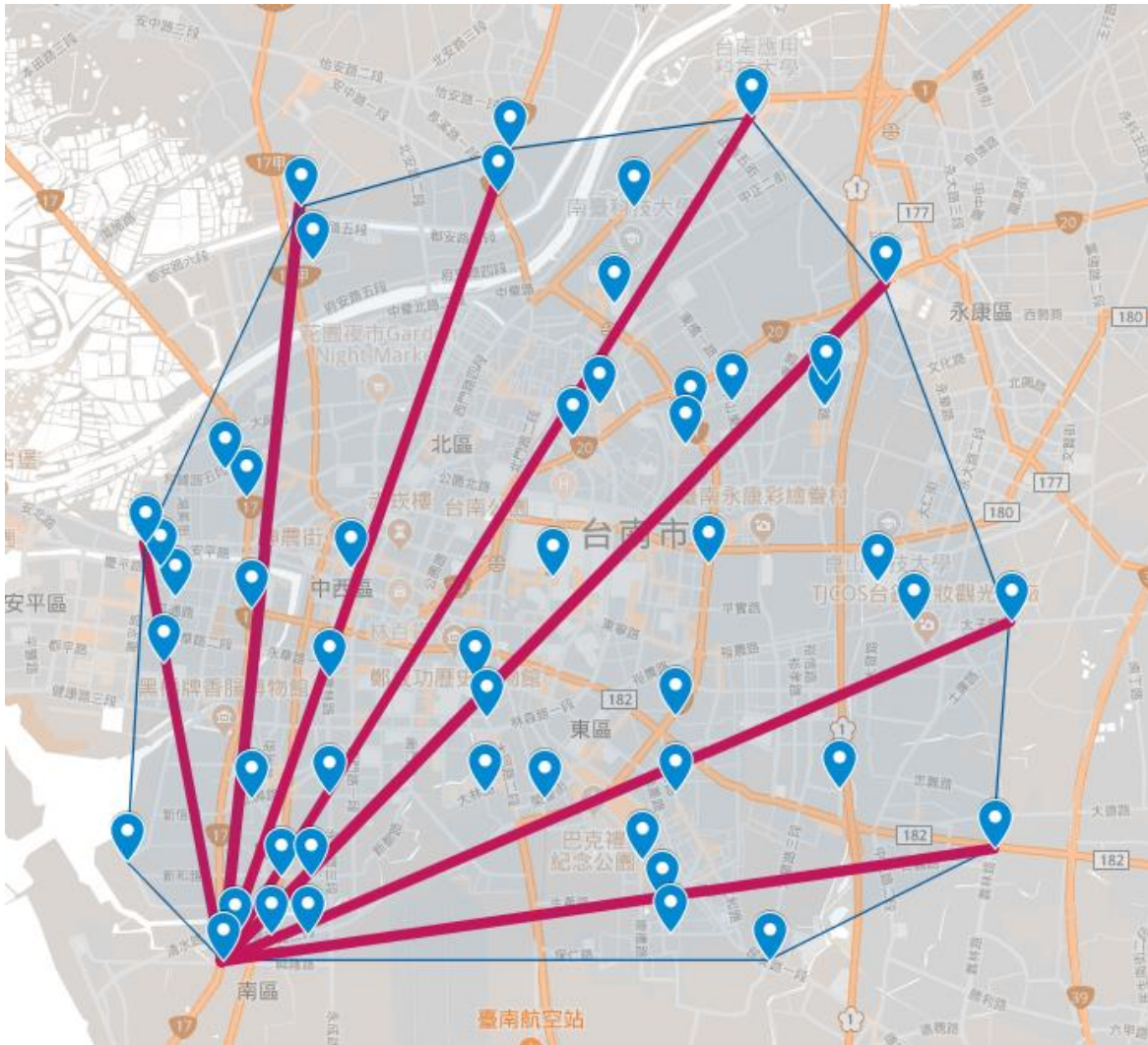


{ 5 }



(2) convex-hull 面積:

把 convex-hull 切成多個三角形，再利用外積算出所有三角形面積，全部加起來就是 convex-hull 的面積了



(3) 最遠距離:

把 convex-hull 邊上的點兩兩組合，算出距離，再找出最長的距離，因為邊上的點都是極點，所以最長的距離一定在邊上

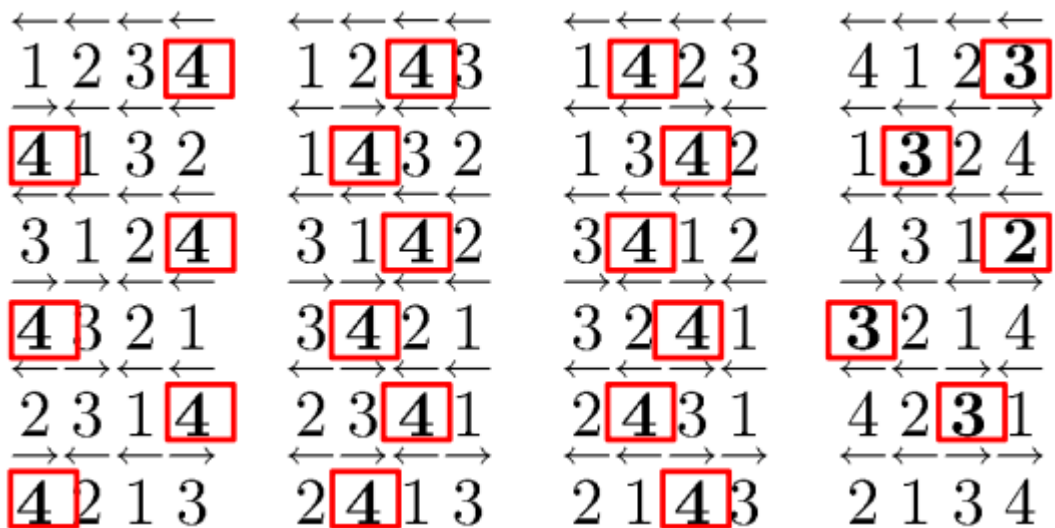
3.

- (a) 假如要到所有  $p$  個 POIs，則最短行程距離是多少？  
(exhaustive search + brute-force algorithm)

把所有可能排列列出來，然後算出每一種排列所需要花的距離，再從裡面找出最小的距離，排列的方法使用 Johnson trotter algorithm，因為這種演算法每次只會和旁邊的點交換，這樣在計算距離時就可以不用全部重新計算，只需要計算有變動的地方。

Johnson trotter algorithm:

1. 初始化所有元素的移動方向向左，輸出自己本身排列
2. 移動最大可以移動的元素(當元素移動方向上的元素比自己小時，才能移動)
3. 反轉所有比移動元素大的元素的移動方向
4. 重複 2~3 步，直到不能移動為止



因為每次只和旁邊的元素交換所以在算距離時只要把交換元素旁邊的距離減掉(綠色箭頭)，然後交換元素，在把交換過後的距離再加回去(紅色箭頭)，如此一來就不用再重新計算全部距離

1 -> 2 -> 3 -> 4 -> 5 -> 1  
1 -> 2 -> 3 -> 5 -> 4 -> 1

(b)假如我們設計一個新演算法：(Convex-Hull-TSP Algorithm)

i. 求出所有點的 Convex-Hull

利用 problem 2 的方法找出 convex-hull

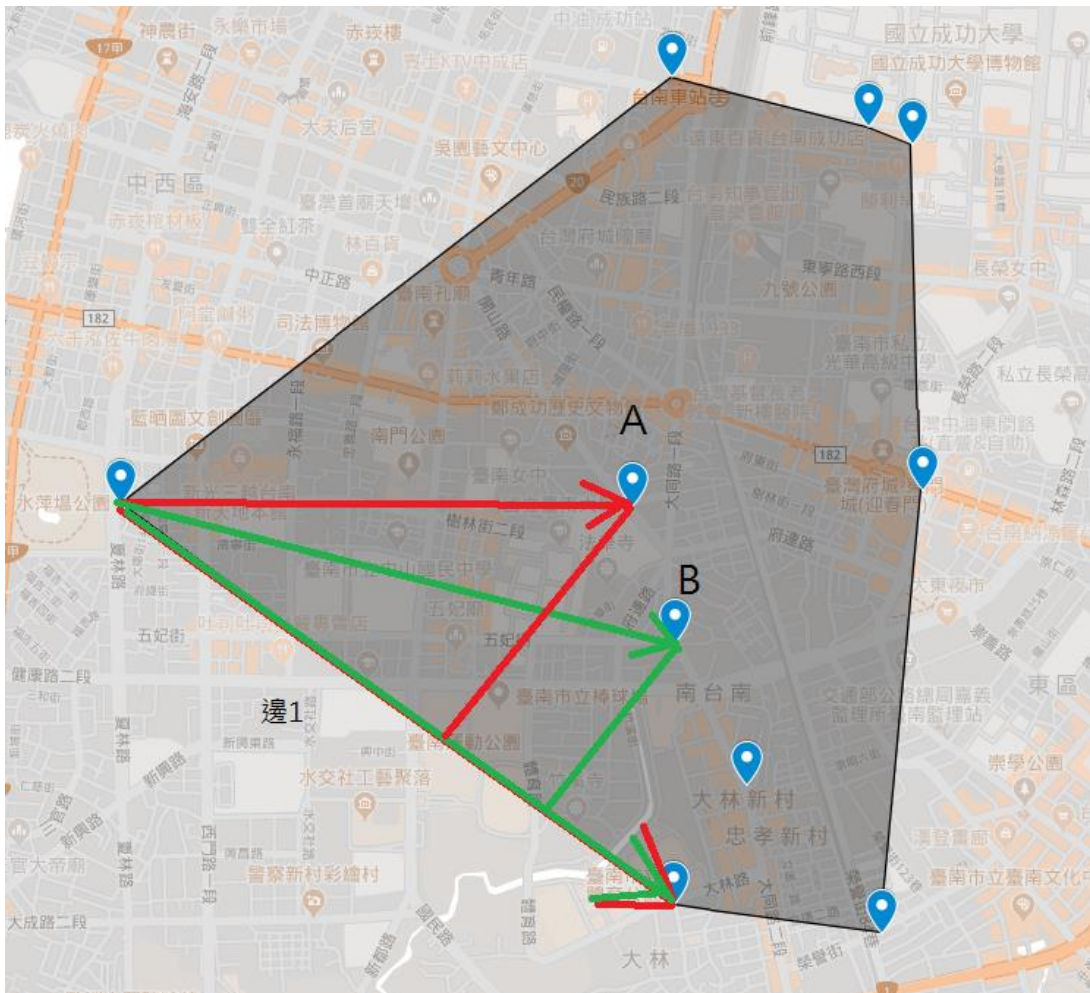
ii. 除 Convex-Hull 上的點外，其餘 POIs 找出離 Convex-Hull 邊最近的點投影

算出在 convex-hull 裡面的每個點對每條邊的距離(距離公式)，把每個點歸類於距離最近的邊上。

iii. 按照投影點依序由 Extreme Point 點開始旅遊，再繞回起始點 請問該演算法的時間複雜度(以 big O 表示)以及(行程)花多少距離？

根據每一條邊計算歸類於這條邊所有點和線的內積(相對於算點到線的起始點距離)，利用內積排序好所有的點在根據線的起始點開始把排序好的點依序連起來最後再連回線的終點。

內積示意圖：





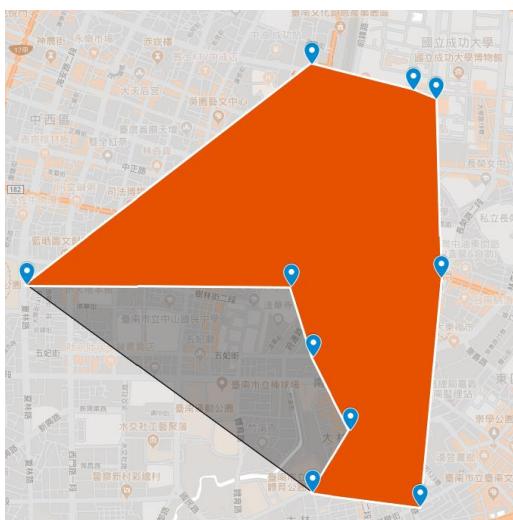
(c)比較 3(a)和 3(b)兩者的距離比較，以及執行時間比較？

距離比較：

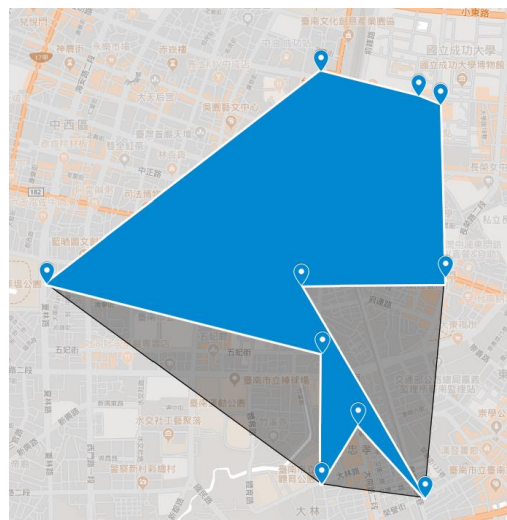
3(a)利用暴力法算出來的距離一定是最短距離，3(b)利用 convex-hull 算出來的距離是相對所有排列組合出來的距離來說是比較短的，只是不一定是最短距離，因為每一個點都只會被歸類於一條線，只能依歸類於那條線的點的投影點去計算距離，因此會忽略到有可能歸類這條線上的某點和歸類於別條線上的點距離相差很近，先連起來會比依照投影點距離來計算還要短。

Example:

最佳路徑



Convex-Hull-TSP Algorithm:



執行時間比較：

3(a)利用暴力法找出來的最佳解所以每個排列都要找出來，所以時間複雜度  $O(n!)$

3(b)先找出 convex-hull 再利用 convex-hell 找出路徑，找出 convex-hull 的時間複雜度為  $O(n^3)$  利用 convex-hull 找出路徑的時間複雜度為  $O(n^2)$  所以 3(b)的時間複雜度為  $O(n^3)$

雖然 3(a)找出來的解是最佳解，但是花費的時間太久了，以真正在使用上，除非點很少，是不會用 3(a)的方法的因為真的花費時間太久，相對來說 3(b)雖然找出來的解不是最佳的但是跑的速度比起 3(a)d 快很多，點少的時候用 3(a)，點多的時候就只能用 3(b)。

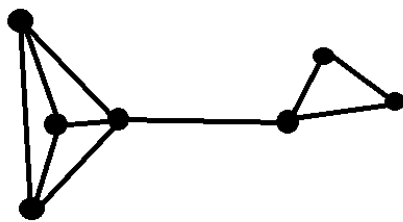
4.

假設我們不可能到全部的 POIs，需要縮短行程，則要找多個 POIs 間的距離不超過一固定範圍( $r$ )形成一個 Group (Clique)  $g$ ，假如 Group  $g$  內的 POIs 個數低於  $k$  個，也不安排到  $g$  的行程。假設符合規範的 Group 共有  $n$  個， $g_i$ ,  $1 \leq i \leq n$ ，列出這  $n$  個 Group 的中心位置  $c_i$ ,  $1 \leq i \leq n$  (或指定該群的任一點 POI or 最靠近的停車場等)。更進一步，我們想要知道假如規劃由一點  $c_j$  開始出發(Group 間開車)，要玩遍所有的 Group 內的 POIs (Group 內走路)，再回到出發處。請問該行程(開車與走路)要花多少距離? (exhaustive search + brute-force algorithms: Clique, TSP problems (TSP 方法用 3(a) or 3(b)亦可，兩者皆實作亦可))

首先把圖上所有點與點的距離小於  $r$  的點全部連起來形成一個新的圖，然後由這張圖的 Poi 個數開始找 group，如果找到 group 就把 group 點從圖上刪除，再從這張圖的 Poi 個數開始找 group 直到找不到 group，最後再選擇從哪個 group 中心位置(該群的任一點 POI)開始 travel 用暴力法找把所有 group 的中心位置和出發點找出最短路徑，再把距離算出來。。

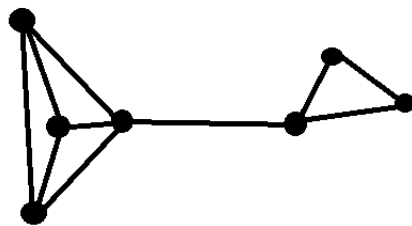
Example:  $k = 3$

(1) 先把圖上點與點的距離小於  $r$  的點全部連起來

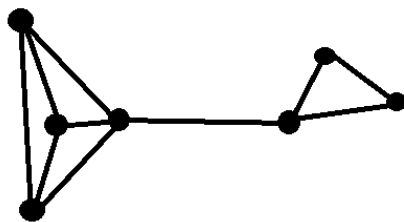


(2) 由這張圖的 Poi 個數開始找 group( $k = 3$ )

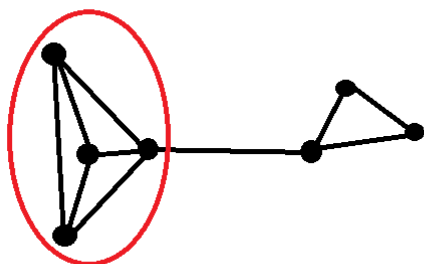
1. 找出 group 大小為 7 沒有完全子圖



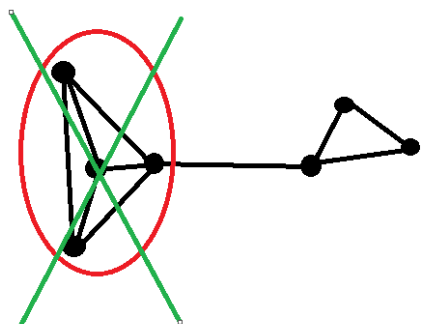
2. 找出 group 大小為 6 沒有完全子圖



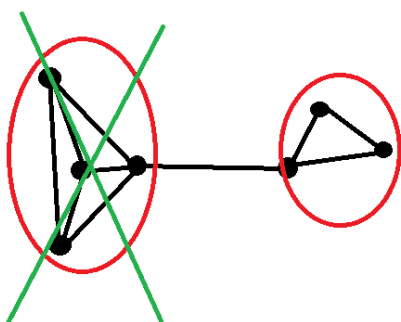
3. 找出 group 大小為 5 沒有完全子圖
4. 找出 group 大小為 4 有完全子圖



5. 把找出來的 group 從圖裡面刪除 代表不會再找一次



6. 找出 group 大小為 3 有完全子圖 因為  $k=3$  所以做到這裡就可以了



7. 選擇出發點 利用暴力法找出出發點和所有 group 中心位置的最短距離，把圖連起來算出距離

### (三) 演算法則

哪兩個 POIs 靠最近？距離多少？

#### Algorithm

//Input: A list P of  $n(n \geq 2)$  points  $P_1=(x_1,y_1) \dots P_n=(x_n,y_n)$

//Output: Indices index 1 and 2 of the closet pair of points

```
dmin  $\leftarrow$   $\infty$ 
for i  $\leftarrow$  1 to n-1 do
  for j  $\leftarrow$  i+1 to n do
    d  $\leftarrow$  sqrt( $(x_i - x_j)^2 + (y_i - y_j)^2$ )
    if d < dmin
      dmin  $\leftarrow$  d;
      index1  $\leftarrow$  i;
      index2  $\leftarrow$  j;
```

時間複雜度(time complexity)  $O(n^2)$

這些 POIs 的範圍有多大 (Convex-Hull，它的面積以及最遠的距離)

#### Algorithm

(a) Convex-Hull

//Input A list P of  $n(n \geq 2)$  points  $P_1=(x_1,y_1) \dots P_n=(x_n,y_n)$

//Output A graph g(use adjacency matrix has n point)

```
for i  $\leftarrow$  1 to n-1 do
  for j  $\leftarrow$  i+1 to n do
    check  $\leftarrow$  0
    for k  $\leftarrow$  1 to n do
      if  $P_k$  is in right side of  $(P_i, P_j)$  then check +1
      else check -1
    if check absolute value equal to n-2 connect  $(P_i, P_j)$  in g
```



(b)面積

```
//Input    A list P of  $n(n \geq 2)$  points  $P_1=(x_1,y_1) \dots P_n=(x_n,y_n)$   
// P are distinct connect points in convex hell  
//Output  graph area
```

```
for i  $\leftarrow$  2 to n do  
    graph area +=  $\langle P_1, P_i \rangle$  Cross product  $\langle P_1, P_{i+1} \rangle$  absolute value (calculate triangle area)
```

(c)最遠的距離

```
//Input A list P of  $n(n \geq 2)$  points  $P_1=(x_1,y_1) \dots P_n=(x_n,y_n)$   
// P are distinct connect points in convex hell  
//Output farthest distance dfar
```

```
dfar  $\leftarrow -\infty$   
for i  $\leftarrow$  1 to n-1 do  
    for j  $\leftarrow$  i+1 to n do  
        d  $\leftarrow$   $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$   
        if d > dfar  
            dfar  $\leftarrow$  d;
```

假如要到所有 p 個 POIs，則最短行程距離是多少？

## Algorithm

```
//Implements Johnson-Trotter algorithm for generating permutations  
//Input    A list P of  $n(n \geq 2)$  points  $P_1=(x_1,y_1) \dots P_n=(x_n,y_n)$   
//Output  min distance of list P
```

```
dmin  $\leftarrow -\infty$ 
```

```
Initialize the first permutation with 1,2...n (all arrows point to left  
While the last permutation has a mobile element do
```

```
    Find its largest mobile element k  
    Swap k and the element of all elements that are larger than k  
    Reverse the direction of all elements that are larger than k  
    if distance in new permutation list > dmin  
        dmin  $\leftarrow$  distance in new permutation list
```

## Algorithm

//Input A list P of  $n(n \geq 2)$  points  $P_1=(x_1,y_1) \dots P_n=(x_n,y_n)$   
 //Output A graph g(use adjacency matrix has n point)

Line\_poi\_pair

```
find convex hull store in g
for all poi not on convex hull
    find closest line
    push poi and closest line to Line_poi_pair
```

```
Sort Line_poi_pair according projection distance and line
for each line in Line_poi_pair
    remove line in g
    connect all pois which is closest to this line and are already sortd
```

假設我們不可能到全部的 POIs，需要縮短行程，則要找多個 POIs 間的距離不超過一固定範圍( $r$ )形成一個 Group (Clique)  $g$ ，假如 Group  $g$  內的 POIs 個數低於  $k$  個，也不安排到  $g$  的行程。假設符合規範的 Group 共有  $n$  個， $g_i$ ,  $1 \leq i \leq n$ ，列出這  $n$  個 Group 的中心位置  $c_i$ ,  $1 \leq i \leq n$  (或指定該群的任一點 POI or 最靠近的停車場等)。更進一步，我們想要知道假如規劃由一點  $c_j$  開始出發(Group 間開車)，要玩遍所有的 Group 內的 POIs (Group 內走路)，再回到出發處。請問該行程(開車與走路)要花多少距離? (exhaustive search + brute-force algorithms: Clique, TSP problems (TSP 方法用 3(a) or 3(b)亦可，兩者皆實作亦可)

## Algorithm

//Input distance range  $R$  and minimum number of poi  $K$  and start travel poi  $C_j$   
 // A list P of  $n(n \geq 2)$  points  $P_1=(x_1,y_1) \dots P_n=(x_n,y_n)$   
 //Output travel distance  $td$

Graph  $g$

Subgraph list sub\_list

Shortest\_path

```
for i ← 1 to n-1 do
    for j ← i+1 to n do
        if distance( $P_i, P_j$ ) <  $R$ 
            connect  $P_i, P_j$  in  $g$ 
number of connect poi in graph n_poi
while n_poi >=  $K$ 
    find complete subgraph in  $g$  which has n_poi number of poi
    if find subgraph
        stroe this subgraph in sub_list
```

```

        delete poi which is found in subgraph from g
        recalculate n_poi in g
    else
        n_poi -= 1
    for each sub_list
        chose center poi

    push Cj to shortest_path
    for each group
        find group shortest path start on center poi
        push group shortest path to shortest_path
    push Cj to shortest_path

    calculate shortest_path distance

```

	時間複雜度	空間複雜度
最近距離	$O(n^2)$	$O(1)$
找出 Convex hull	$O(n^3)$	$O(n^2)_$
Convex hull 面積	$O(n)$	$O(1)_$
最遠距離	$O(n^2)$	$O(1)_$
最短行程(brute force)	$O(n!)$	$O(1)_$
Convexhull TSP Algorithm	$O(n^3)$	$O(n)$
Problem4 Group 路徑	$O(2^{k-1})$ k is connect poi number	$O(n^2)$

## (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 Code + MinGw64(gcc)

### 3. 電腦硬體

CPU: Intel i5-9400f

Main Memory: 16GB

Disk: 1T HDD

500G SSD

250G SSD

OS: W10 x64



## (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 (圖是由 adjacency matrix 儲存)

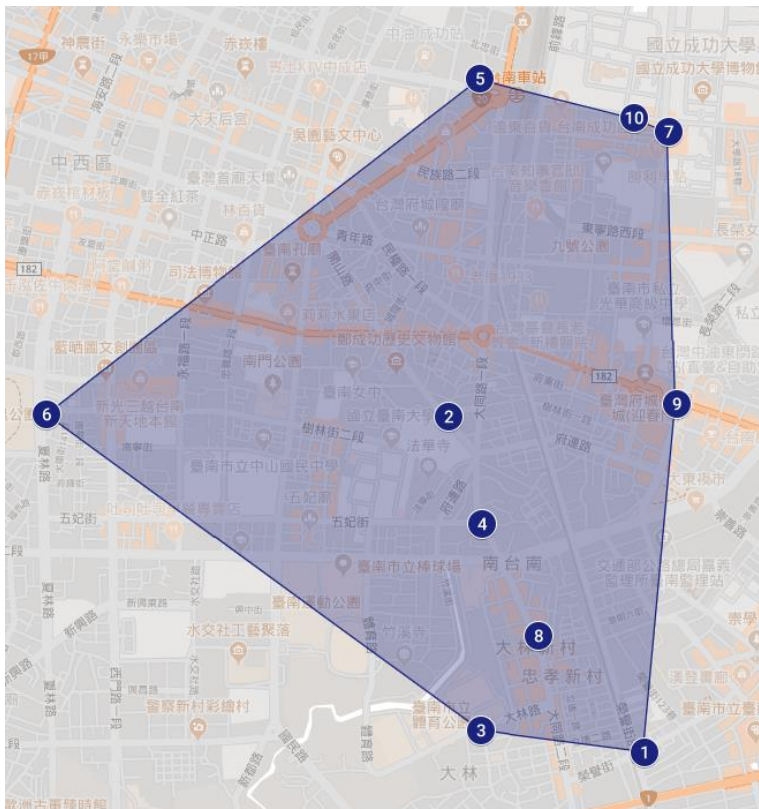
輸入檔案名稱

```
Enter file name.....map.in
```

POI 個數和地點資訊和參照圖

(經緯度會轉換成平面座標地圖以下算出來的距離和實際上的誤差不會差很多)

```
10
(1) 7-ELEVEN 榮譽門市,120.2173976,22.9746374
(2) 7-ELEVEN 開山門市,120.2100878,22.9861404
(3) 7-ELEVEN 少觀門市,120.2112832,22.975369
(4) 7-ELEVEN 大五門市,120.2113108,22.9824398
(5) 7-ELEVEN 龍成門市,120.2112536,22.9976986
(6) 7-ELEVEN 公道門市,120.1950887,22.9862257
(7) 7-ELEVEN 育樂門市,120.2182848,22.9958866
(8) 7-ELEVEN 億載門市,120.2134439,22.9786426
(9) 7-ELEVEN 東龍門市,120.2185615,22.9865827
(10) 7-ELEVEN 新東育門市,120.2169895,22.9963772
```



第一題:找最短距離

```
-----
7-ELEVEN 育樂門市 1與7-ELEVEN 新東育門市距離最近 距離為:0.143486 km
執行時間: 0.0027 ms
-----
```

第二題:找出 convex hull 和最遠距離和面積

```
-----
convex hull start ....
(7) 7-ELEVEN 育樂門市
|
(10) 7-ELEVEN 新東育門市
|
(5) 7-ELEVEN 龍成門市
|
(6) 7-ELEVEN 公道門市
|
(3) 7-ELEVEN 少觀門市
|
(1) 7-ELEVEN 榮譽門市
|
(9) 7-ELEVEN 東龍門市
|
(7) 7-ELEVEN 育樂門市
7-ELEVEN 龍成門市 與7-ELEVEN 榮譽門市 距離最遠
最遠距離為: 2.63045 km
Convex hull 面積為: 3.77019 km^2
-----
```

第三題:先輸入最短距離起始點位置，輸出最短距離

```
-----
Enter start poi to find shortest path....
-----
```

exhaustive search + brute-force algorithm

```
-----
exhaustive search + brute-force algorithm
(3) 7-ELEVEN 少觀門市
|
(8) 7-ELEVEN 億載門市
|
(4) 7-ELEVEN 大五門市
|
(2) 7-ELEVEN 開山門市
|
(6) 7-ELEVEN 公道門市
|
(5) 7-ELEVEN 龍成門市
|
(10) 7-ELEVEN 新東育門市
|
(7) 7-ELEVEN 育樂門市
|
(9) 7-ELEVEN 東龍門市
|
(1) 7-ELEVEN 榮譽門市
|
(3) 7-ELEVEN 少觀門市
最短距離為: 8.69443 km
花費時間: 36.0083 ms
-----
```

## Convex-Hull-TSP Algorithm

```
-----
Convex-Hull-TSP Algorithm
(3) 7-ELEVEN 少觀門市
|
(8) 7-ELEVEN 億載門市
|
(1) 7-ELEVEN 榮譽門市
|
(2) 7-ELEVEN 開山門市
|
(9) 7-ELEVEN 東龍門市
|
(7) 7-ELEVEN 育樂門市
|
(10) 7-ELEVEN 新東育門市
|
(5) 7-ELEVEN 龍成門市
|
(6) 7-ELEVEN 公道門市
|
(4) 7-ELEVEN 大五門市
|
(3) 7-ELEVEN 少觀門市
距離為: 9.74106 km
所花費時間0.0713 ms
-----
```

第四題: 輸入範圍 R 和數量 K 輸出每一個 group

```
Enter range R ....1.5
Enter group poi number K .....3
```

```
Group 1 center: (1) 7-ELEVEN 榮譽門市
(1) 7-ELEVEN 榮譽門市
|
(2) 7-ELEVEN 開山門市
|
(3) 7-ELEVEN 少觀門市
|
(4) 7-ELEVEN 大五門市
|
(8) 7-ELEVEN 億載門市
|
(9) 7-ELEVEN 東龍門市
|
(1) 7-ELEVEN 榮譽門市
Group 2 center: (5) 7-ELEVEN 龍成門市
(5) 7-ELEVEN 龍成門市
|
(7) 7-ELEVEN 育樂門市
|
(10) 7-ELEVEN 新東育門市
|
(5) 7-ELEVEN 龍成門市
```

輸入起始點位址(group center )

輸出旅遊路徑

```
Which group center you want to start travel.....1
路程為:
(1) 7-ELEVEN 榮譽門市
|
(3) 7-ELEVEN 少觀門市
|
(8) 7-ELEVEN 億載門市
|
(4) 7-ELEVEN 大五門市
|
(2) 7-ELEVEN 開山門市
|
(9) 7-ELEVEN 東龍門市
|
(1) 7-ELEVEN 榮譽門市
|
(5) 7-ELEVEN 龍成門市
|
(7) 7-ELEVEN 育樂門市
|
(10) 7-ELEVEN 新東育門市
|
(5) 7-ELEVEN 龍成門市
|
(1) 7-ELEVEN 榮譽門市
總路徑長為: 10.9168 km
```

## 2. Input Code Format.



```
map.in - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
10
7-ELEVEN 榮譽門市,120.2173976,22.9746374
7-ELEVEN 開山門市,120.2100878,22.9861404
7-ELEVEN 少觀門市,120.2112832,22.975369
7-ELEVEN 大五門市,120.2113108,22.9824398
7-ELEVEN 龍成門市,120.2112536,22.9976986
7-ELEVEN 公道門市,120.1950887,22.9862257
7-ELEVEN 育樂門市,120.2182848,22.9958866
7-ELEVEN 億載門市,120.2134439,22.9786426
7-ELEVEN 東龍門市,120.2185615,22.9865827
7-ELEVEN 新東育門市,120.2169895,22.9963772
```

第一列為 資料數目 接下來每一列都是由地點名稱，經度，緯度所組成的紀錄，每一列不同屬性由逗號分隔

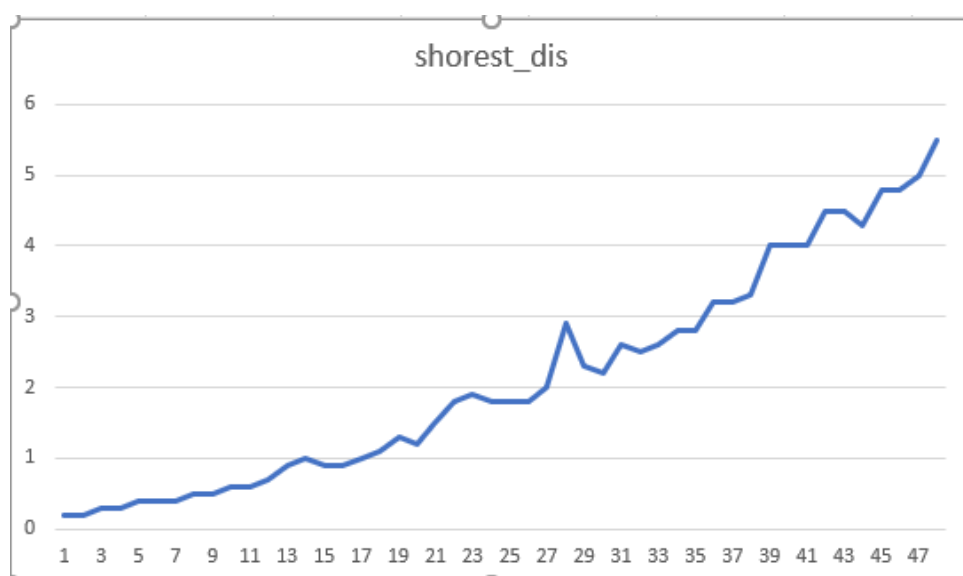


## (五) 執行結果、討論與心得

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

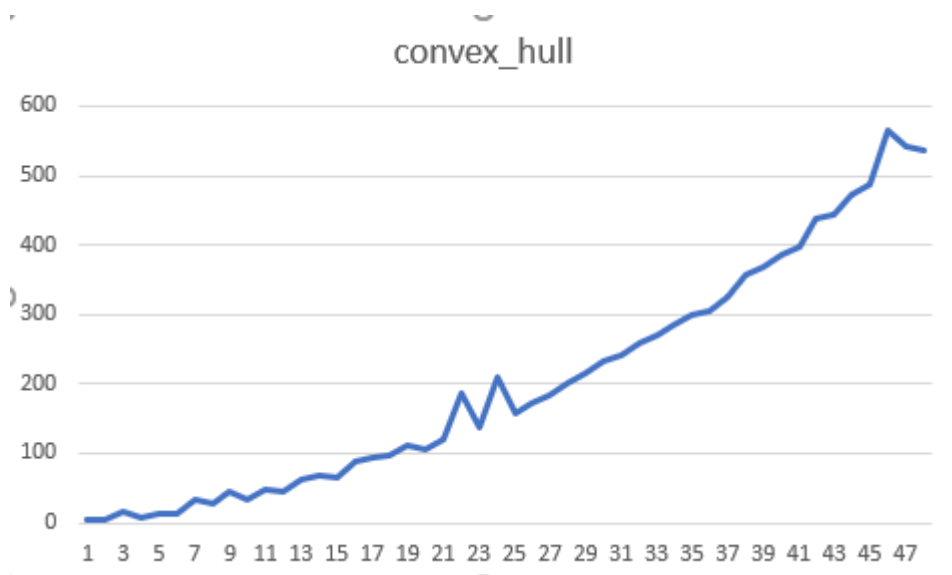
#### (1) 找出最短距離



POI 個數與時間(ms) 關係圖

因為最短距離  $O(n^2)$  所以當數量上升時間也會跟著上升

#### (2) 找出 convex hull

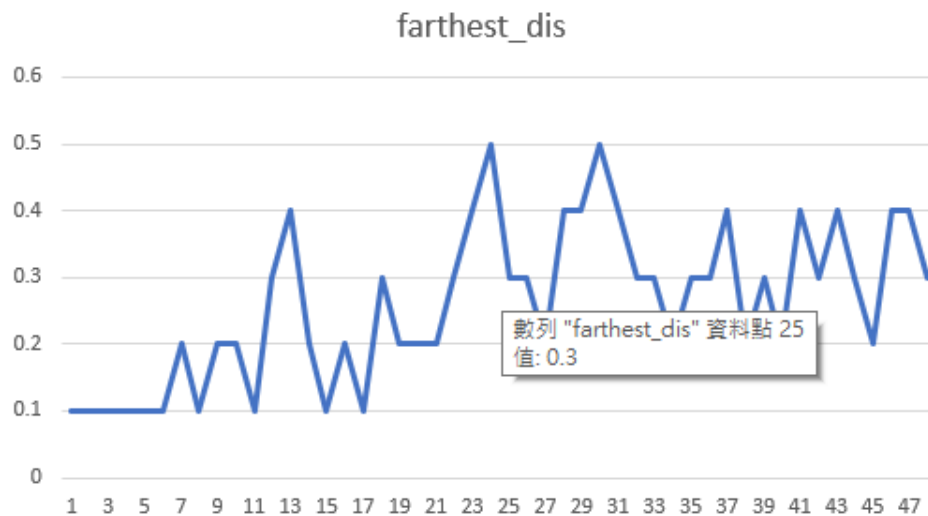


POI 個數與時間(ms) 關係圖

因為  $O(n^3)$  所以當數量上升時間也會跟著上升只是因為樣本數量不太夠所以

上升的比例不太明顯

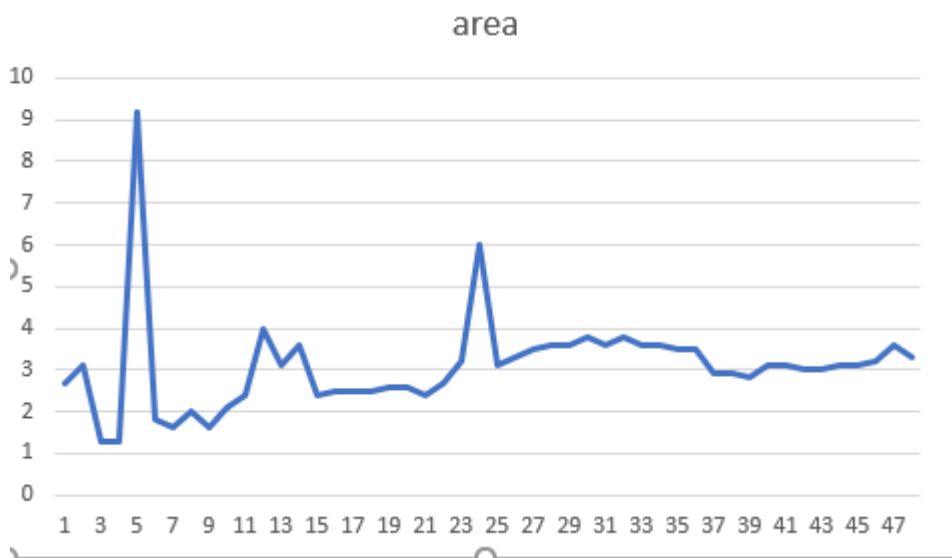
### (3) 找出最遠距離



POI 個數與時間(ms) 關係圖

因為最遠距離會根據 convex hull 的點的個數來找的所以每個樣本點的個數都不一樣所以會有起伏出現，

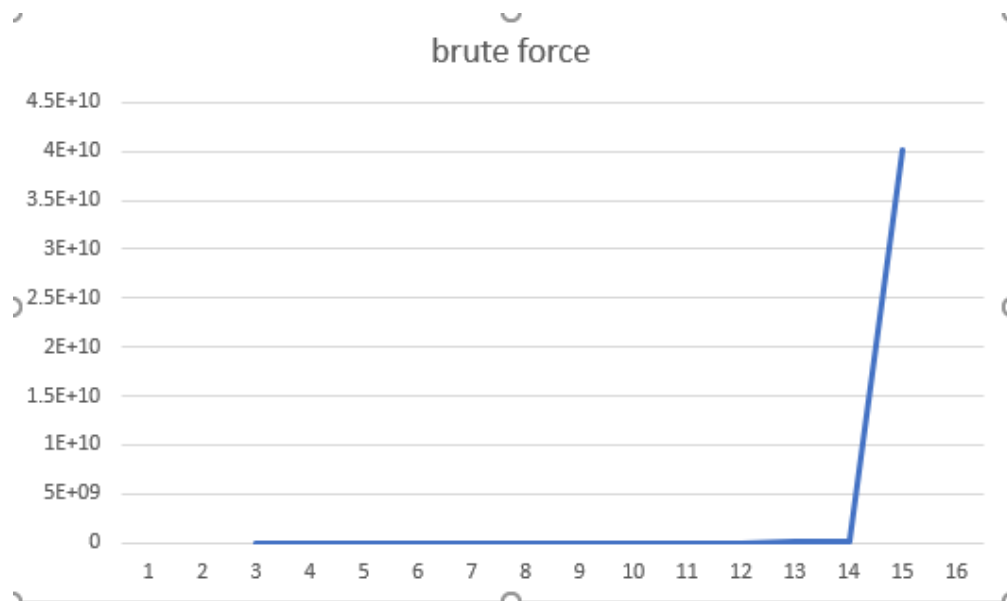
### (4) 找出 convex hull 面積



POI 個數與時間(ms) 關係圖

因為最面積會根據 convex hull 的邊的個數來計算的所以每個樣本點的個數都不一樣所以會有起伏出現

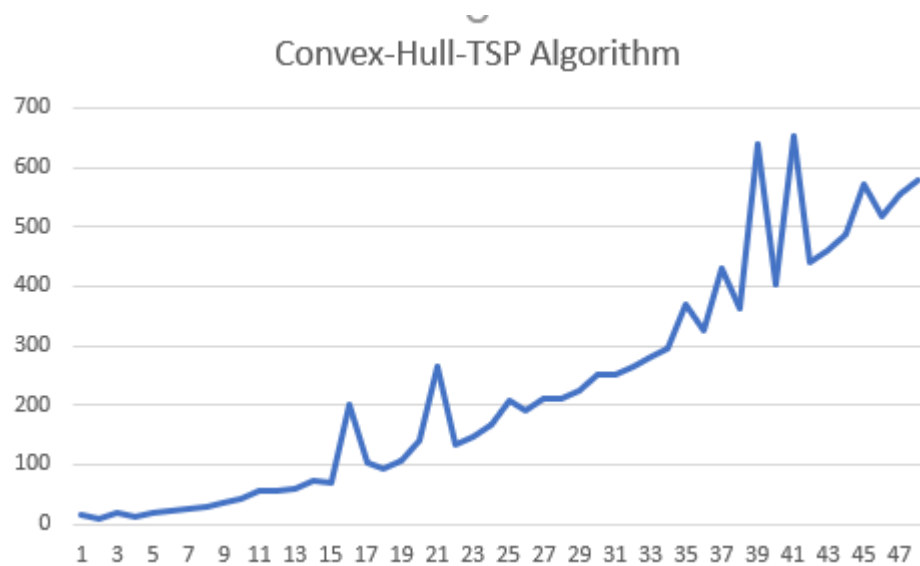
### (5) 找出最短路徑(brute force)



POI 個數與時間(ns) 關係圖

最短路徑是用暴力法所以  $O(n!)$ ，當 POI 個數少的時候時間很短，但是成長速度極快所以到後面會時間會上升非常快只是因為後面上升幅度太大所以前面幾個樣本上升的幅度相對來說微不足道

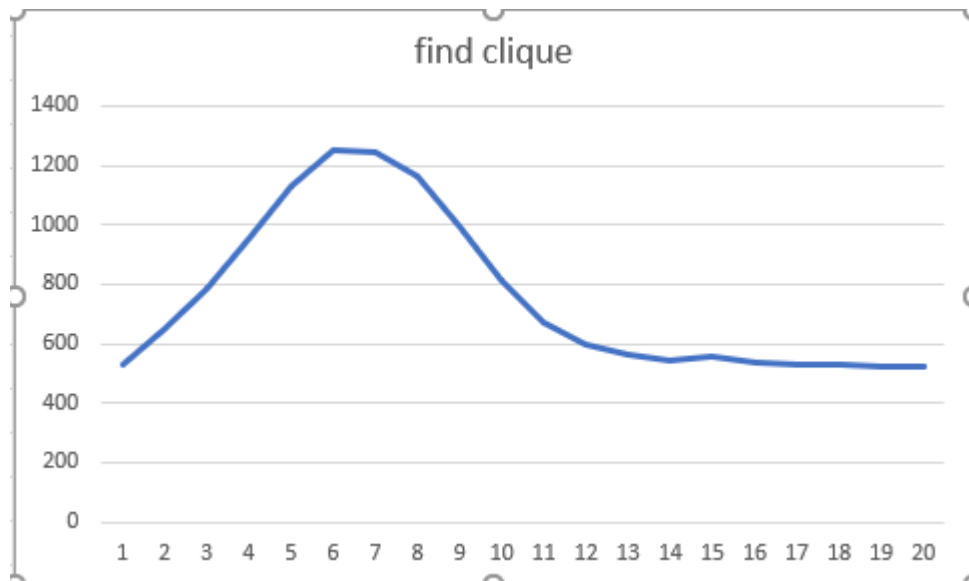
### (6) 找出最短路徑( Convex-hull-TSP Algorithm)



POI 個數與時間(ns) 關係圖

Convex-hull-TSP Algorithm 的時間複雜度為  $O(n^3)$  和 convex-hull 一樣，只是因為採樣不夠多所以看起來沒有上升很快

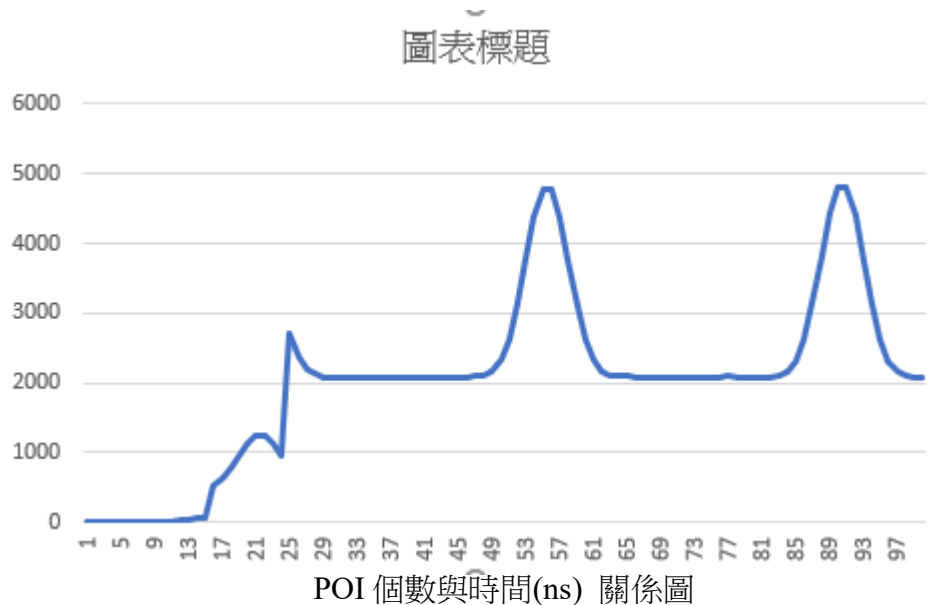
(7) Problem4 當點的個數為固定時( $n=23, r=5$ )  $k$  與時間的變化量



K 與時間(ns) 關係圖

當  $k$  小的時候需要找出的 group 數較少，當  $k$  往中間靠時可以組合的個數變多所以時間相對來說就會上升了， $k$  繼續上升需要尋找的 group 就變少了就下降了

(8) Problem4 當點的個數為固定時( $n=23, k=5$ )  $r$  與時間的變化量



POI 個數與時間(ns) 關係圖

當  $r$  上升時有可能 graph 上相連的 poi 增加 也有可能不變，poi 數目增加時間就會上升不變的話就不會增加，至於突出的部分可能是因為 cpu 排程問題所以當時這個程式優先權被放到後面所以忽然上升。



## 2. 心得與討論

這次的實驗讓我有寫演算法的感覺，第三題的演算法不同所需要花的時間就差很多了，雖然不一定找得出最佳解但是以時間上來說，是相對可以接受的，因為找出來的解是比較接近最佳解的，而最佳解花的時間太久了這次光是跑第三題的 a 測試 poi 為 16 個的時候就覺得好像跑不完了，可是換演算法跑到 50 個 poi 都沒問題還是很快，讓我了解到寫程式演算法的重要性。

## 參考文獻

[1]

- [1] Anderson-Freed Introduction to the design & analysis of algorithms / Anany Levitin
- [2] Fundamentals Of Data Structures in C 2<sup>nd</sup> Edition / Horowitz · Sahni & Anderson-Freed
- [3] GPS 经纬度坐标转平面坐标的简化计算方法及精度分析\_肖体琼
- [4] <https://www.studyblue.com/notes/n/cmpsc-465-midterm-2/deck/21768075>
- [5] 台南大學 E-course 演算法課程 Algorithms Report Format
- [6] <https://noob.tw/data-structure/>