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 | ： | 林星宇 |
| 學號 | ： | S10659029 |

老師：陳宗禧

中華民國 108年10月30日

內容

[**壹、** **簡介及問題描述** 3](#_Toc23343052)

[1. 簡介 3](#_Toc23343053)

[2. 問題 3](#_Toc23343054)

[**貳、** **理論分析** 4](#_Toc23343055)

[1. Brute-Force algorithm 4](#_Toc23343056)

[2. Convex-Hull brute-force algorithm 4](#_Toc23343057)

[3. Exhaustive search + brute-force algorithm 4](#_Toc23343058)

[4. Exhaustive search + brute-force algorithms 4](#_Toc23343059)

[**參、** **演算法則** 5](#_Toc23343060)

[1. Brute-force algorithm 5](#_Toc23343061)

[2. Convex-Hull brute-force algorithm 5](#_Toc23343062)

[3. Exhaustive search + brute-force algorithm 5](#_Toc23343063)

[4. Exhaustive search + brute-force algorithms 5](#_Toc23343064)

[**肆、** **程式設計環境架構** 6](#_Toc23343065)

[1. 程式語言 6](#_Toc23343066)

[2. 程式開發工具 6](#_Toc23343067)

[3. 電腦硬體 6](#_Toc23343068)

[**伍、** **程式 (含source code, input code, and output code)** 7](#_Toc23343069)

[1. 主程式 7](#_Toc23343070)

[2. Input Code Format 10](#_Toc23343071)

[3. Output Code Format 10](#_Toc23343072)

[**陸、** **執行結果、討論與心得** 11](#_Toc23343073)

[1. 執行結果 11](#_Toc23343074)

[2. 討論 12](#_Toc23343075)

[3. 心得 13](#_Toc23343076)

[**柒、** **參考文獻** 13](#_Toc23343077)

1. **簡介及問題描述**

設計與實作Complete Graph 判斷，理論驗證與實驗分析該問題!

1. 簡介

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. 問題
2. 哪兩個 POIs 靠最近? 距離多少? (brute-force algorithm)
3. 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. 假如要到所有 p 個 POIs，則最短行程距離是多少? (exhaustive search + brute-force algorithm)
   2. 假如我們設計一個新演算法：(Convex-Hull-TSP Algorithm)
4. 求出所有點的 Convex-Hull
5. 除 Convex-Hull 上的點外，其餘 POIs 找出離 Convex-Hull 邊最近的點投影
6. 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?
   1. 比較 3(a)和 3(b)兩者的距離比較，以及執行時間比較?
7. 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，gi, 1 ≤ 𝑖 ≤ 𝑛，列出這n個Group 的中心位置ci, 1 ≤ 𝑖 ≤ 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cj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))
8. **理論分析**
9. Brute-Force algorithm

**for** i 🡨 0 **to** n – 1 **do**

**for** j 🡨 i + 1 **to** n **do**

**if** distance(A[ i, j]) < temp

temp = distance(A[ i, j])

1. Convex-Hull brute-force algorithm
2. 先求出兩點(x1, y1), (x2, y2)形成的線y = ax + b
3. 利用Brute-force 藉由y – ax > b 或 y – ax < b 判斷每點在線的左邊或右邊
4. 如果每點皆在同側，則形成此線段之兩點為Convex-Hull 邊上的點
5. Exhaustive search + brute-force algorithm

ShortestTravel(int a, int b, int\* num) {

**if** (a == b)

**for** i 🡨 0 **to** n **do**

temp = num[i];

**if** (comparedistance > distance)

comparedistance = distance;}

**else**

**for** j 🡨a **to** n **do**

**swap**(num[a], num[j]);

**ShortestTravel**(a + 1, b, num);

**swap**(num[a], num[j]);

}

1. Exhaustive search + brute-force algorithms
2. 輸入至少經過n點，最常路徑r
3. 利用組合列出所有路徑可能，並計算路徑
4. 比較是否超過r並顯示結果
5. **演算法則**
6. Brute-force algorithm

利用兩層for迴圈算出兩點之間的距離並存取距離最小之值

1. **演算法時間複雜度(time complexity)**

O(n^2)

1. **演算法空間複雜度(space complexity)**

O(1)

1. Convex-Hull brute-force algorithm

利用兩層for迴圈求出兩點直線方程式，並比較所有點在此直線之同側或異側

1. **演算法時間複雜度(time complexity)**

O(n^2)

1. **演算法空間複雜度(space complexity)**

O(1)

1. Exhaustive search + brute-force algorithm

利用recursive排列所有可能並記錄各排列總距離

1. **演算法時間複雜度(time complexity)**

O(n!)

1. **演算法空間複雜度(space complexity)**

O(1)

1. Exhaustive search + brute-force algorithms

利用recursive求出組合後計算重心

1. **演算法時間複雜度(time complexity)**

O(n!)

1. **演算法空間複雜度(space complexity)**

O(1)

1. **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

1. 程式開發工具

Visual Studio 2019

1. 電腦硬體

作業系統: Window10 家用版

系統類型: 64位元作業系統，x64型處理器

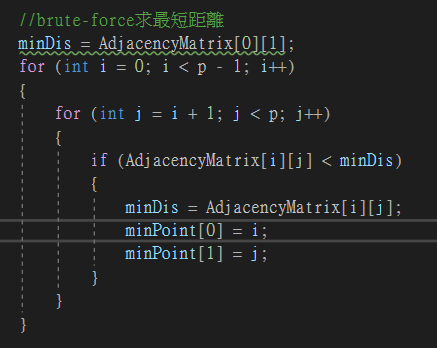
處理器: Intel® Core™ i7-7700HQ CPU @ 2.80GHz

顯示卡: Geforce® GTX 1050 Ti with 4GB GDDR5 Graphics

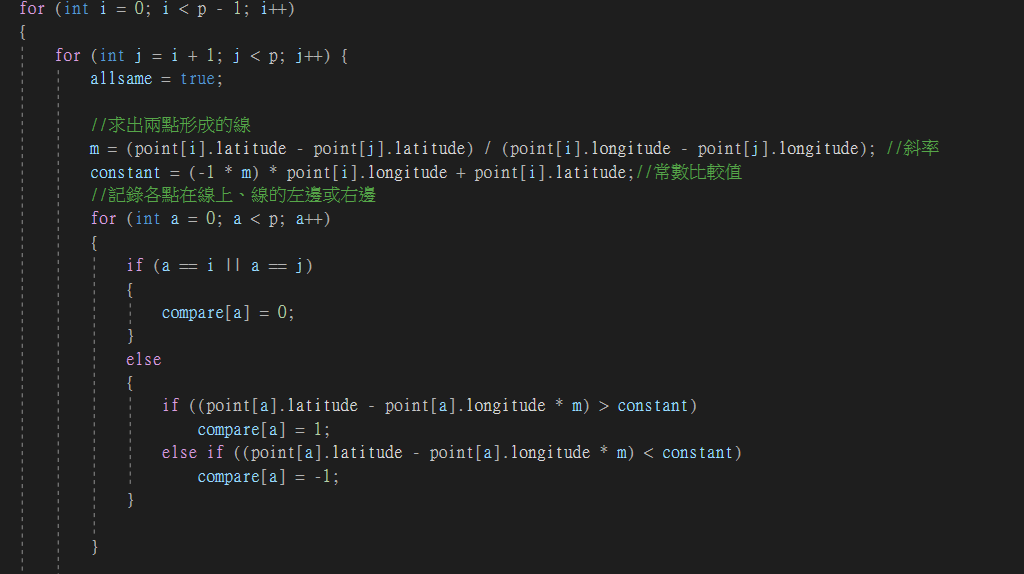
1. **程式 (含source code, input code, and output code)**

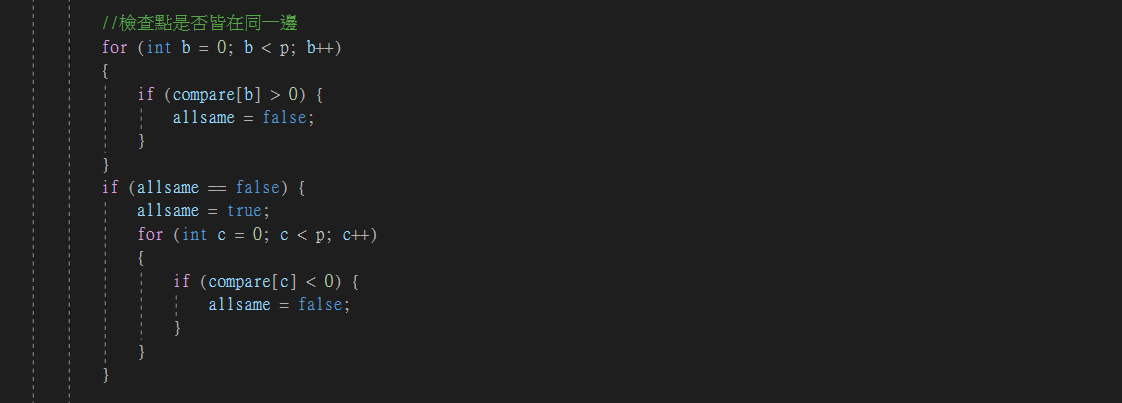
程式含source code, input code, and output code等…

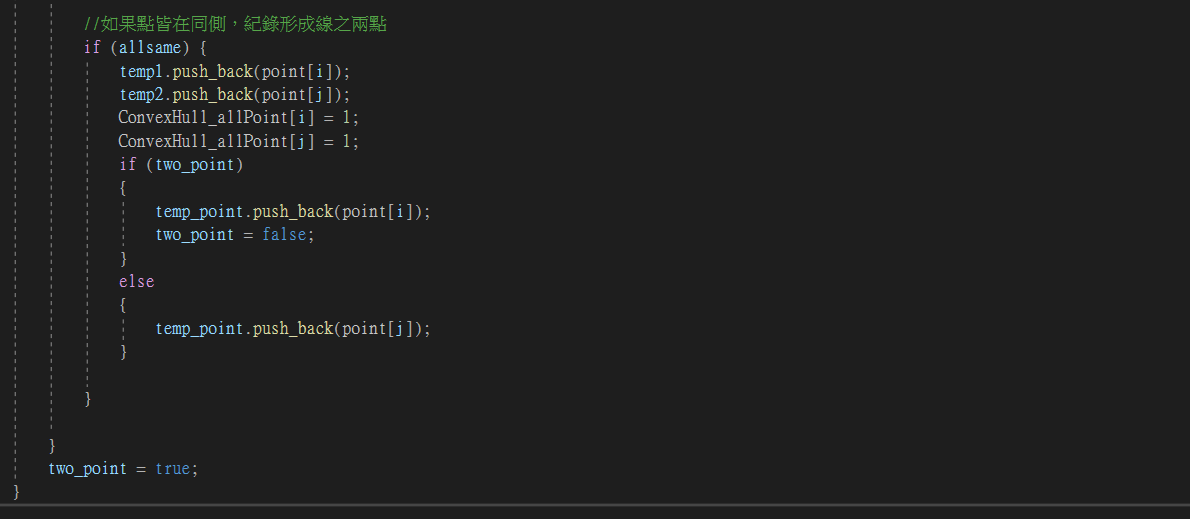
1. 主程式
2. Brute-Force algorithm:



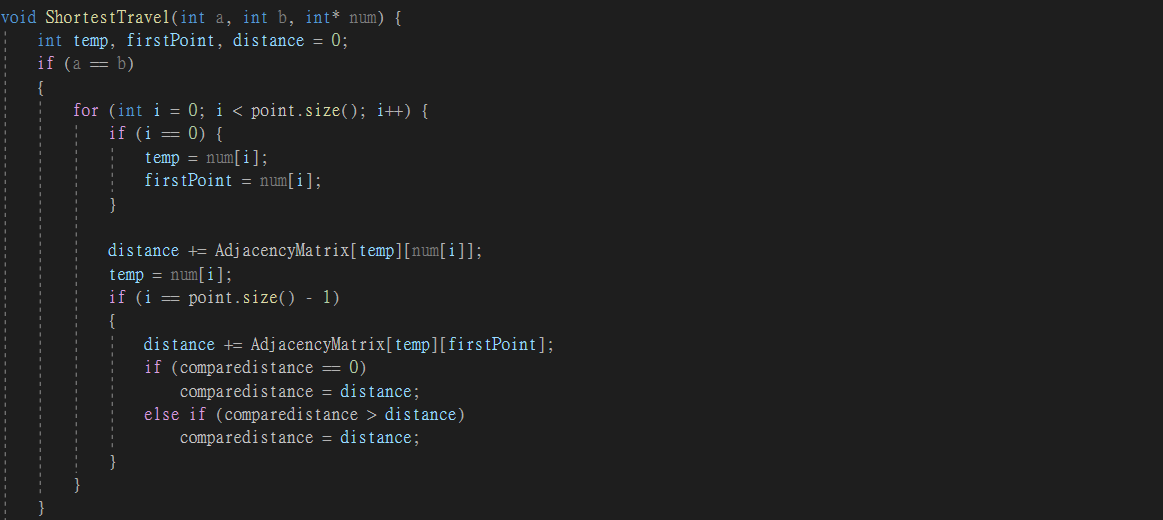
1. Convex-Hull brute-force algorithm





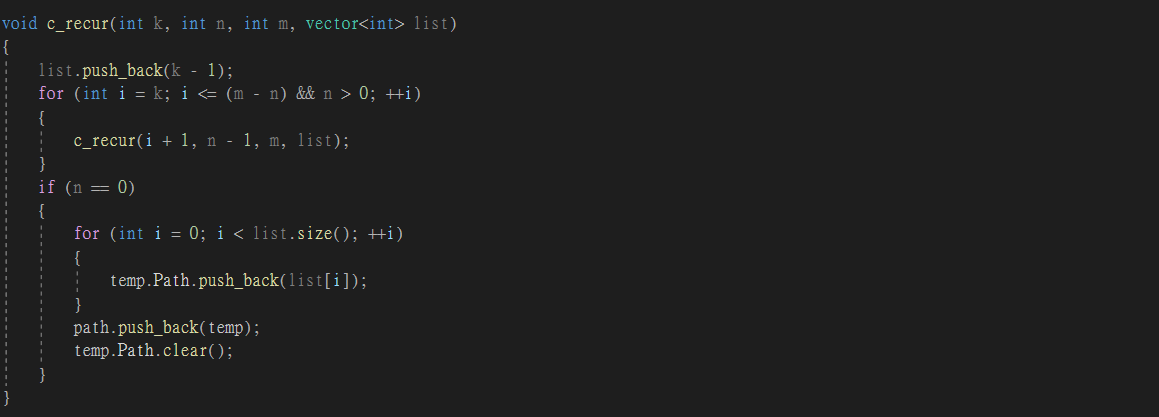


1. Exhaustive search + brute-force algorithm

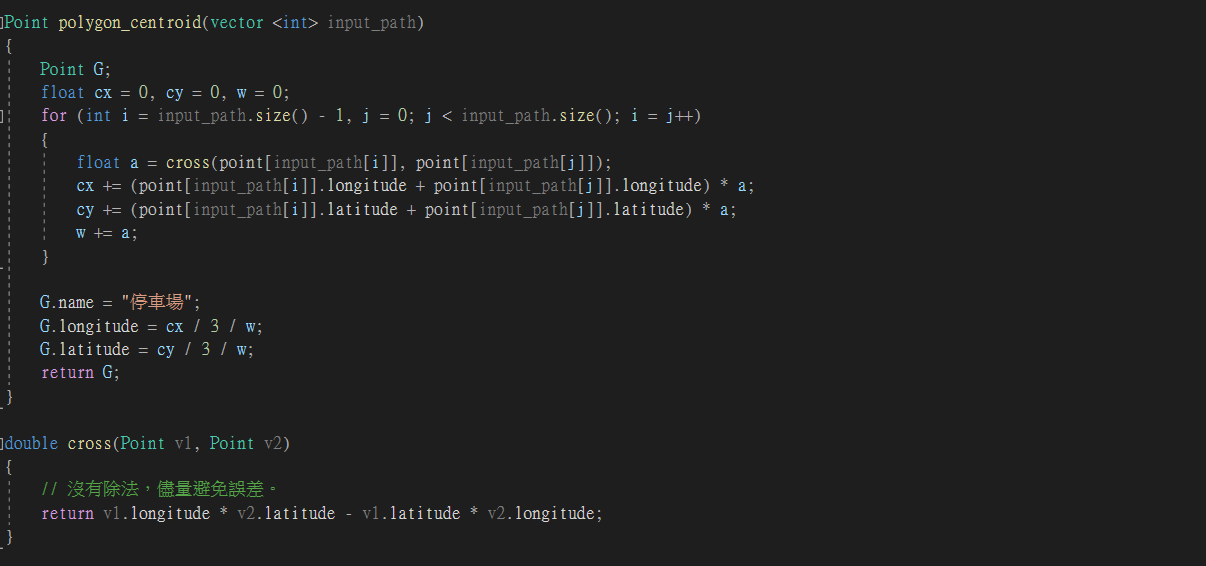




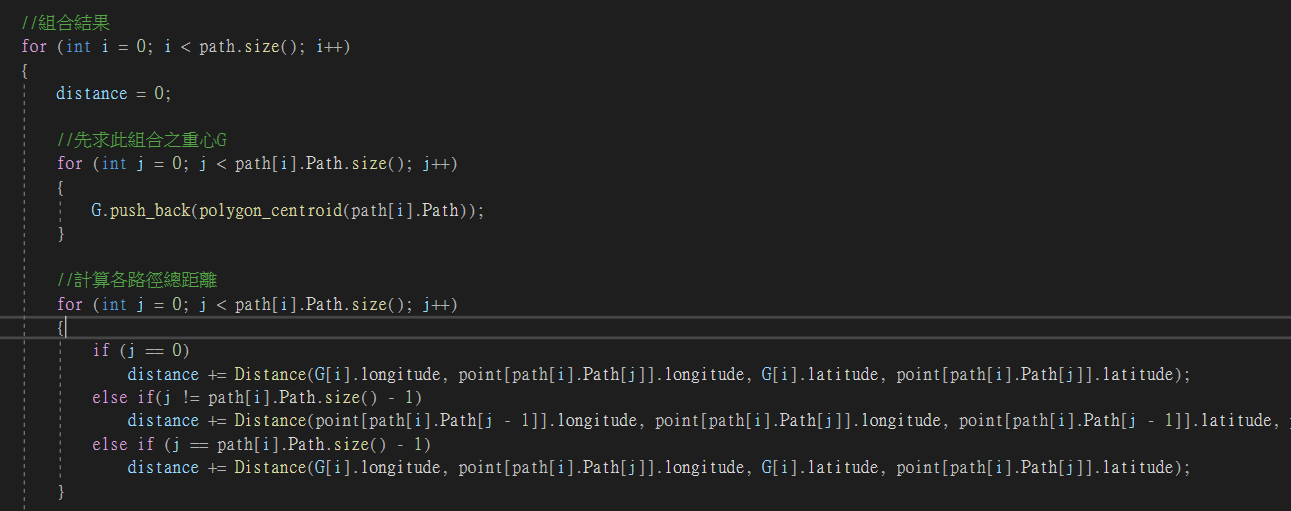
1. Exhaustive search + brute-force algorithms
2. 做組合C(n, p)

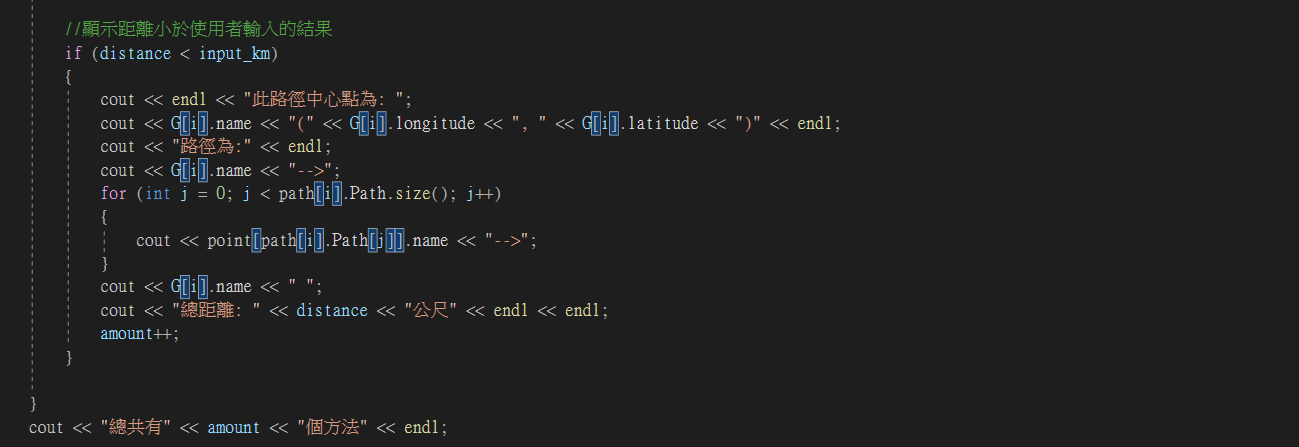


1. 求出路徑之重心



1. 計算路徑和比較





1. Input Code Format

Three of examples for input use are in below….

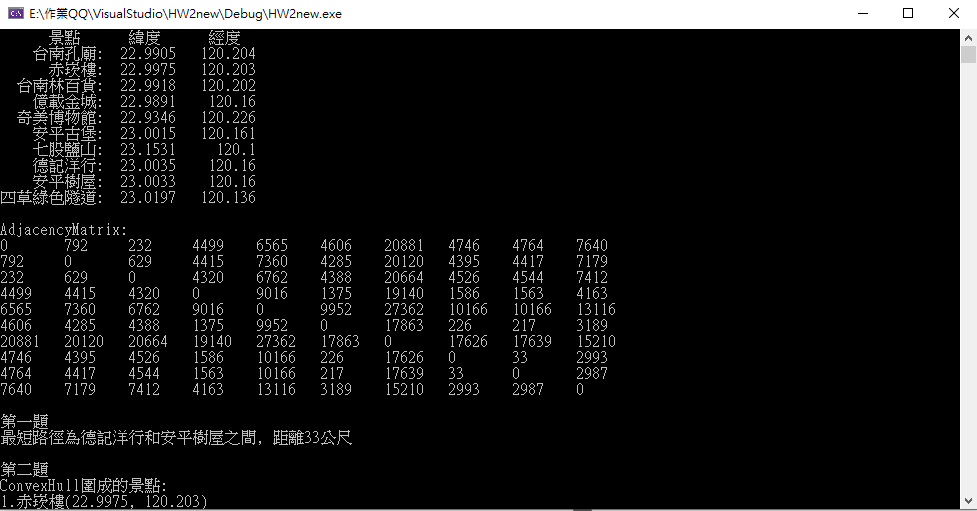
1. input1.txt : 7個點
2. input2.txt : 10個點
3. input3.txt : 11個點
4. Output Code Format
5. 景點、緯度、經度
6. TSP(以 Adjacency Matrix 表示)
7. 第一題
8. 第二題
9. Convex Hull 圍成的景點
10. Convex Hull 圍成最大面積
11. Convex Hull 最遠兩點
12. 第三題
13. Brute-Force 最短距離
14. Convex Hull TSP
15. 比較兩者距離差距
16. 第四題
17. 路徑
18. 方法個數
19. **執行結果、討論與心得**

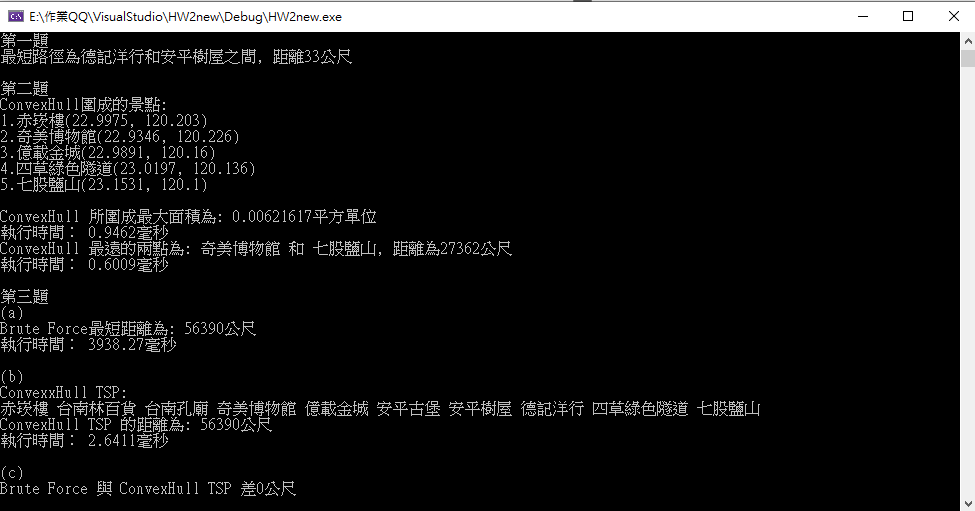
執行結果與討論 (執行時間、problem *n*的大小等問題討論)等…

1. 執行結果

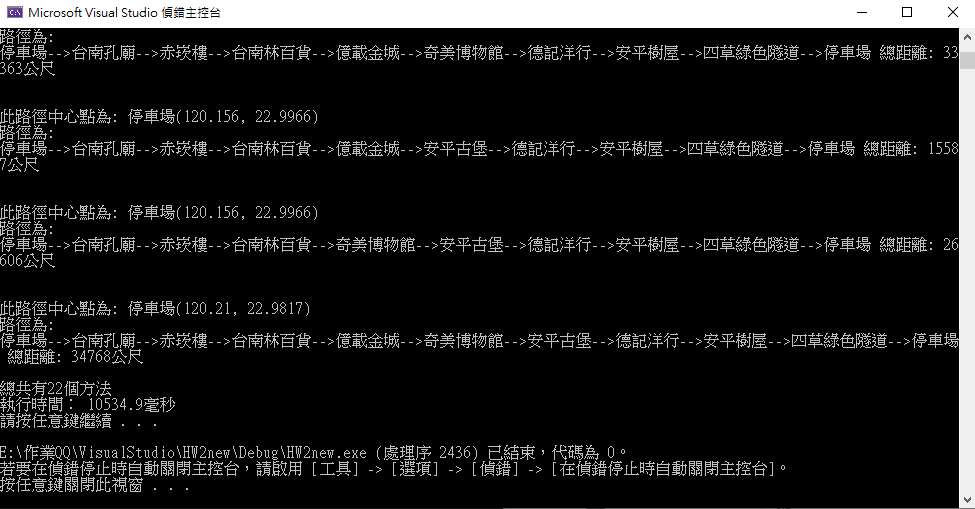
Output of program:

以10個點為例:









1. 討論

執行時間、問題大小等問題討論! 利用 MS Excel 畫出問題大小與執行時間的關係!

Running Time (單位:毫秒)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 2.面積 | 2.最遠距離 | 3.(a)最短距離 | 3.(b)Convex Hull TSP | 4.至少5點，距離<30000公尺 |
| 7個點 | 0.4596 | 1.0188 | 6.8152 | 2.8256 | 50378.4 |
| 10個點 | 0.6832 | 0.7581 | 3837.05 | 6.3134 | 39472.9 |
| 11個點 | 1.5438 | 0.778 | 42652.4 | 3.5204 | 48586.3 |

1. 心得

從圖表結果可以看出，最短距離因為使用Brute-Force，O(n!)，此演算法非常耗時，當點少變多時，所需時間的上升速度非常快，因此3(b)採用的方法雖然未必是最佳解，但以空間換取時間有時是很重要的。

1. **參考文獻**
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms," Third Edition, The MIT Press, 2009.
3. R.C.T. Lee, S.S. Tseng, R.C. Chang, and Y.T.Tsai, "Introduction to the Design and Analysis of Algorithms," McGraw-Hill, 2005.
4. Anany V. Levitin, "Introduction to the Design and Analysis of Algorithms," 2nd Edition, Addison Wesley, 2007.
5. Richard Neapolitan and Kumarss Naimipour, "Foundations of Algorithms," Fourth Edition, Jones and Bartlett Publishers, 2010.