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: Eletric Vehicle Motor**

|  |  |  |
| --- | --- | --- |
| 班級 | ： | 資工三 |
| 姓名 | ： | 陳武雋 |
| 學號 | ： | S10959020 |

老師：陳宗禧

中華民國 111年10月31日

# 目錄

1. **簡介及問題描述摘要….……………..…………………………………………1**
   1. **簡介…………….…………………………………………………………………………2**
   2. **問題………………………….……………………………………………………………4**
2. **理論分析….………………………..………………………………………8**
3. **演算法則….…………………………..……………………………………10**
4. **程式設計環境架構.………………………..…………………………………12**
5. **程式.…………………………………………..………………………………14**
6. **執行結果、討論與心得.………………………..……………………………18**

參考文獻……………………………………………………………………………22

**(一) 簡介及問題描述摘要**

指派n個充電站給m台電動機車，共有五個相關問題

1. 簡介

求最靠近的充電站和電動機車以及距離，再求充電站的Convex-hull範圍(面積及最遠距離)，又設n=m且每個充電站分配一個空位，如何配對讓耗能最小，最後若分配的空位不只一個

2. 問題

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. m 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 n=m 且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)
6. 如(5)所述，假設 n≠m，第 i 個充電站空位有𝑣𝑖個, 𝑣𝑖 ≥ 0, 1 ≤ 𝑖 ≤ 𝑚，如何解決這個 問題? (Hungarian method or Exhaustive Search)

**(二) 理論分析**

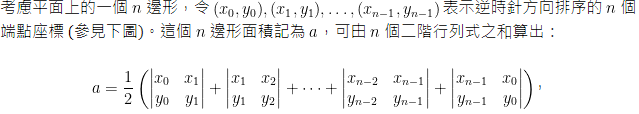
1. **Find closest point distance (Brute-force)**

距離公式: 根號(兩點x座標相減之平方+兩點y座標相減之平方)，用兩層迴圈算出全部充電站或電動車的兩點距離，存出最小距離，距離為0的點是重複點，不算成最小距離。

1. **Find extreme point of convex-hull (Brute-force)**

兩點算方程式ax+by-c=0，a = y1 -y2，b = x2- x1，c= y2 \* x1 – y1\* x2，給兩點找方程式要先計算兩點斜率m = (y2-y1)/ (x2-x1)，再找過直線上一點假設找x1,y1的已知點，方程式: y – y1 = m (x- x1)把斜率換回座標的表示也就是y – y1 = (y2-y1)/ (x2-x1)\*(x- x1)，移項: (y-y1)\*(x2-x1) – (y2-y1)\*(x-x1) = 0，乘開會是 y\*x2 -y\*x1 -y1\*x2+y1\*x1-y2\*x+y2\*x1+y1\*x-y1\*x1 = 0，把能消去的 y1\*x1消掉+結合率 y(x2-x1)+x(y1-y2) – y1\*x2 + y2\*x1 = 0，就是方程式ax+by+c=0，把所有點帶入看結果是> < = 0，如果全部是>或< 0的答案，說明兩點形成的線就是convex-hull的邊，全部這些點就會形成convex-hull。

1. **Convex-hull area**

不規則形面積 = 1/2 \* 行列式，參考文獻5則有證明網址，注意點要是相鄰點，並且結尾需再補上選定的第一個點之x、y 

1. **Find Convex-hull點中最遠 point distance (相對於1)**

相對於找closest point，找出Convex-hull中最大距離。

1. **Hungarian method**

Step1先形成充電站和電動車間的距離矩陣，每1距離就1耗能，使用第一題之距離公式。

Step2 選出每列最小值，把那一列減去最小值，再換成欄同樣減去最小值。

Step3 找出只含0的最佳指派，若找不到往Step4

Step4 打勾再畫線

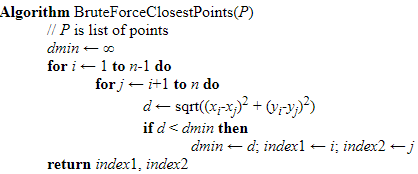
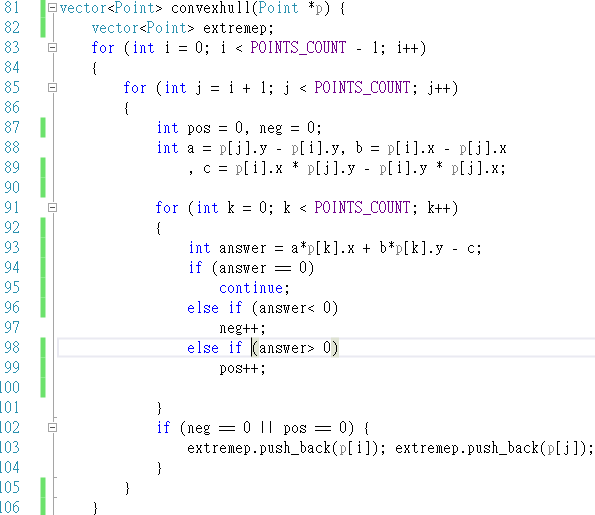
打勾:

* 1. 對沒有獨立0的行打勾
  2. 對打勾的行含0的列
  3. 對打勾列含獨立0元素的行打勾
  4. 重複直到 2. 3.沒新的勾出現

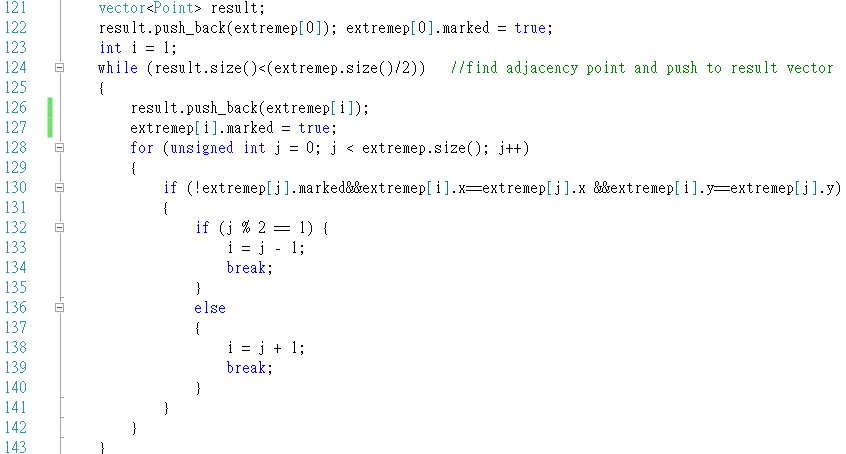
畫線: 對沒打勾的列和打勾的行畫線

Step5 找出沒被打勾元素中的最小值，沒畫線的列減去最小值，若造成負值把畫線的行加回最小值，回Step3重複到最佳指派

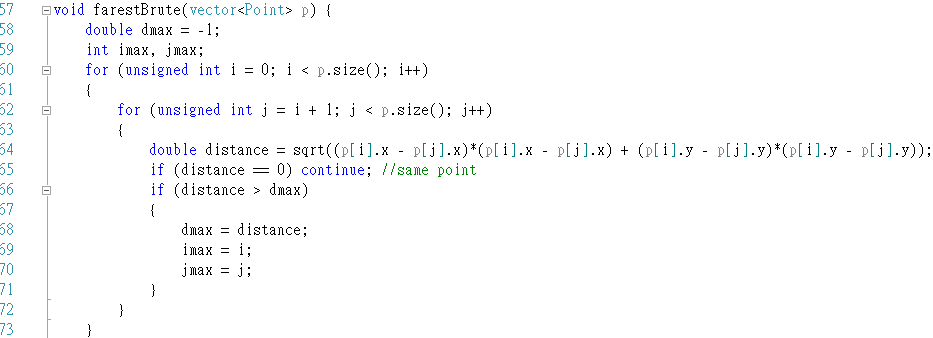
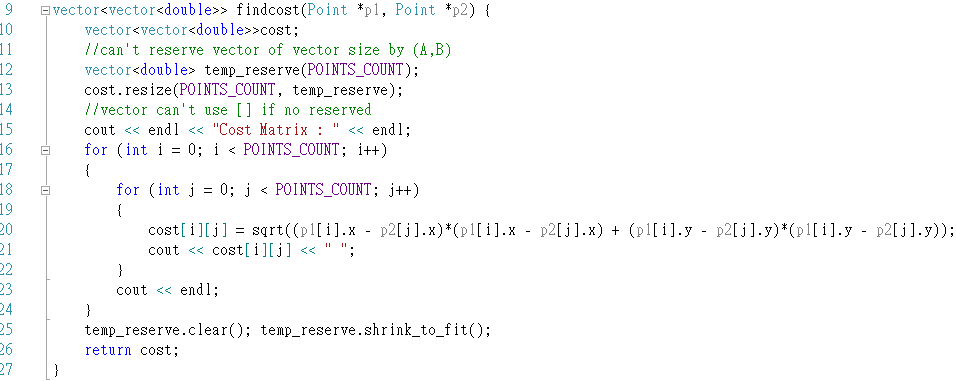
**(三) 演算法則**

1. 時間複雜度是: O(n^2)，basic operation是平方，屬於n(n-1)/2的問題，算一次距離做兩次平方，共做n(n-1)次。空間複雜度是:O(1)
2. 時間複雜度是: O(n^3)，空間複雜度是: O(n)。

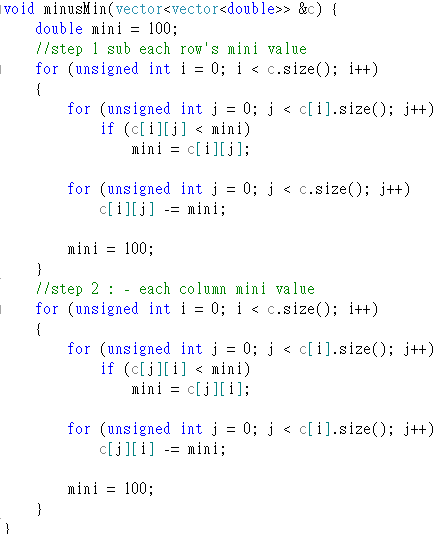
basic operation是比較大小，次數是n\*n(n-1)/2，每次方程式比較正負做一次比較\*1。空間複雜度是2n個點，Class Point則存x、y座標，所以空間複雜度該要算O((2n)\*2)。

1. 時間複雜度: O(n^2)，空間複雜度: O(n)。

Basic operation是加減法assignment，共做了n\*(n/2)次，每次分配索引值做一次加減法assignment，空間複雜度是n個點，同上一題存x、y座標，所以算O(n\*2)

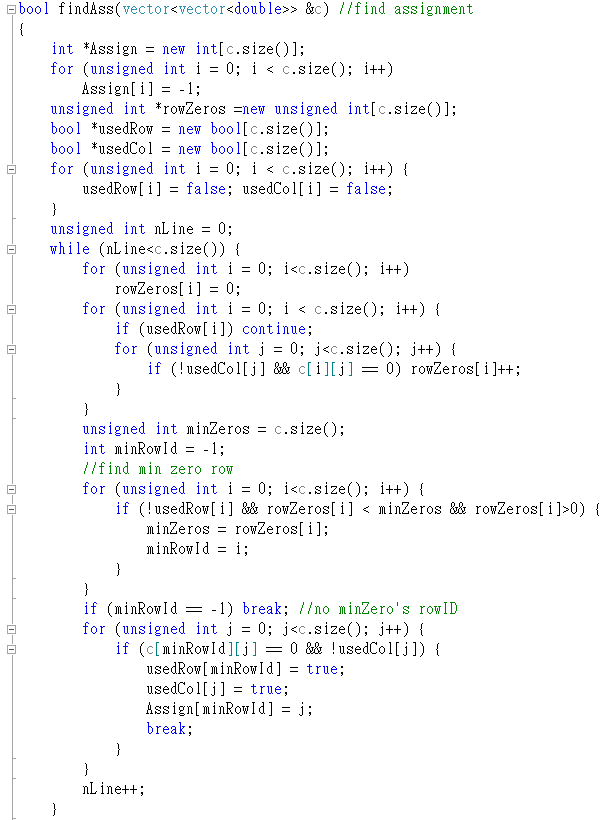
1. 相對於最短距離，時間複雜度也為O(n^2)，空間複雜度是O(1)。
2. 找出耗能的二維向量，時間複雜度為O(n^2)，空間複雜度為O(n^2)

Basic operation是乘法每次算距離都會做兩次，和上面最遠/短距離演算法一樣時間複雜度，空間複雜度是存起來的二維向量 n^2。

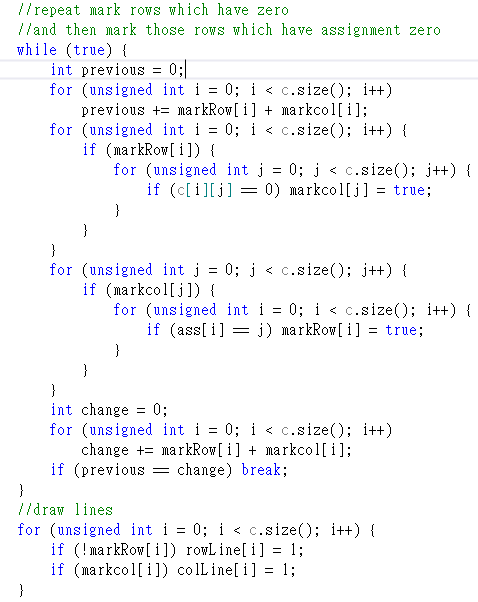


找出每行每列的最小值再該行/列減去最小值，時間複雜度為O(n^2)，空間複雜度是O(1)。

Basic operation是比大小，需要做n^2次，每次也只比一個大小\*1。

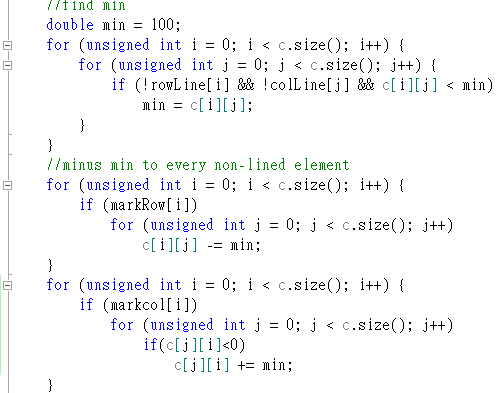
找出最佳指派，時間複雜度O(n)，空間複雜度O(n)

Basic operation是assign行路徑的索引至Assign陣列，每次該行沒在儲存used行的陣列中且是最小0元素的列中的0，做三次assign，共3\*n次。額外用不少大小n的陣列，但空間複雜度為O(n)。



打勾和畫線，時間複雜度O(n)，空間複雜度O(n)。

需要不少額外大小n的陣列來存mark和畫的線，但仍為O(n)，basic operation為判斷是否為標記過的列和是否為標記的直行，需做n次每次兩個判斷，共2\*n，仍為O(n)。



最後每列沒被畫線的減去全部沒被畫線的最小值，若有負值則要在該直行補回減去的最小值，時間複雜度為O(n^2)，空間複雜度O(1)。

Basic operation為減法，做了n^2次每次做一次減法，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 2015

3. 電腦硬體

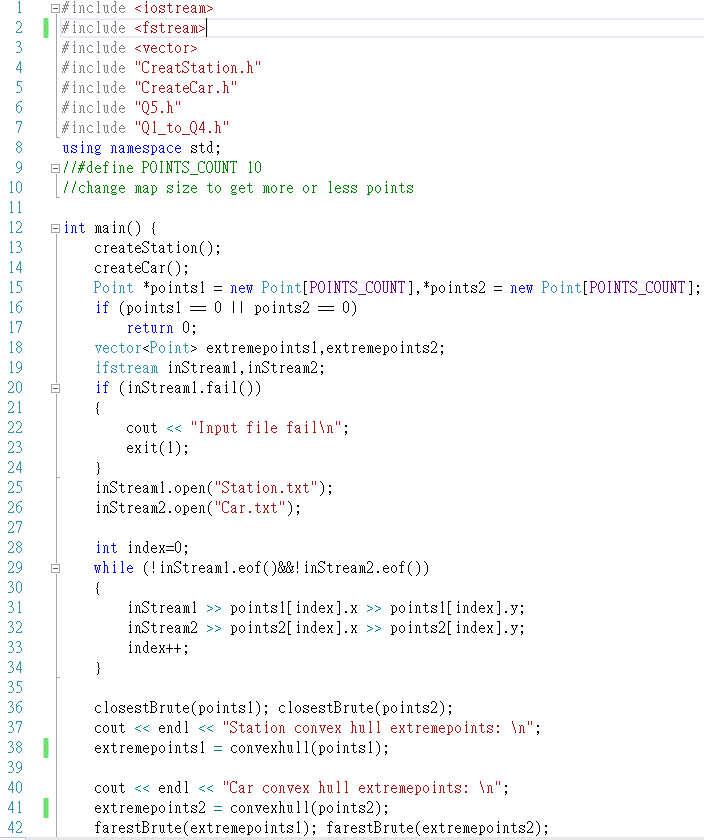
CPU: Intel I5, Main Memory: 16GB

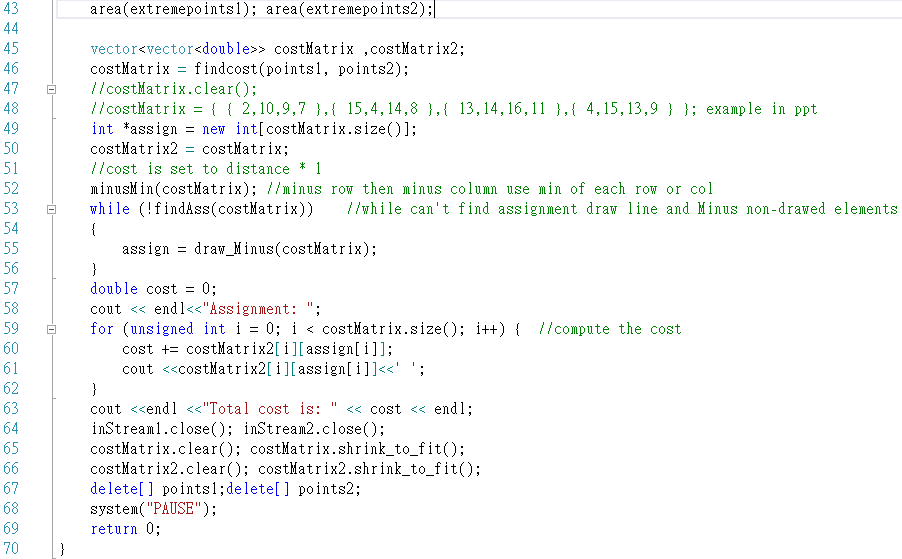
**(五) 程式 (含source code, input code, and output code)**

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

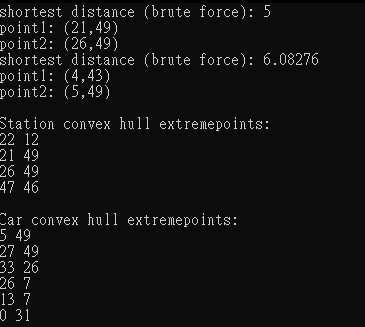
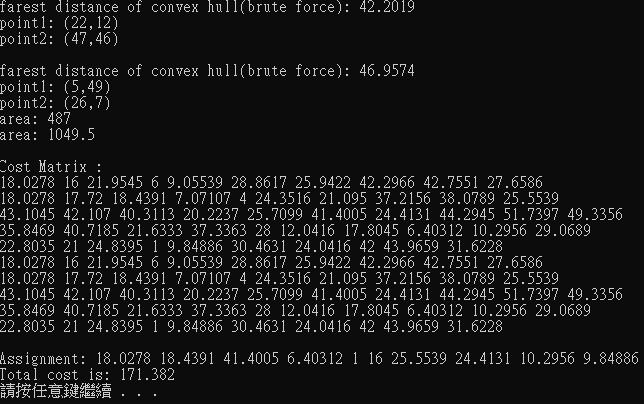
1. 主程式

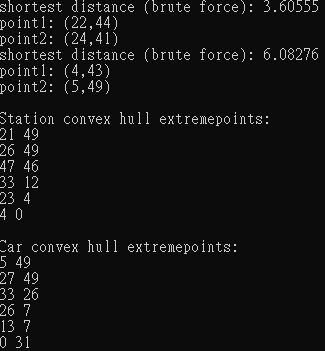
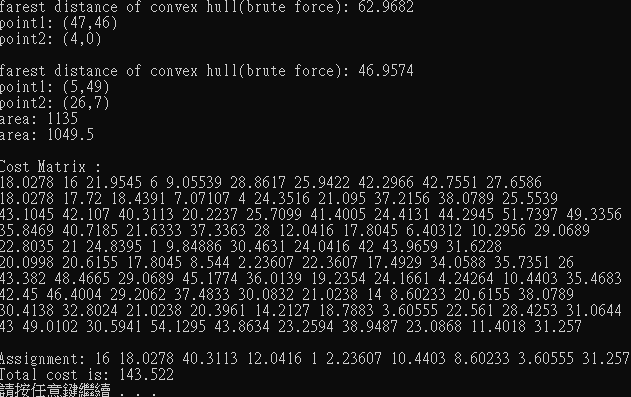
C program





2. Output Code Format

第六題的輸出結果: 

第五題的輸出結果:  

Car.txt (不會改動)

|  |  |
| --- | --- |
| 4 | 43 |
| 5 | 49 |
| 10 | 30 |
| 27 | 49 |
| 22 | 40 |
| 14 | 21 |
| 33 | 26 |
| 26 | 7 |
| 13 | 7 |
| 0 | 31 |

Station.txt (充電站容量K =1)

|  |  |
| --- | --- |
| 21 | 49 |
| 22 | 44 |
| 47 | 46 |
| 22 | 12 |
| 26 | 49 |
| 24 | 41 |
| 23 | 4 |
| 33 | 12 |
| 31 | 29 |
| 4 | 0 |

Station.txt(充電站容量K=2)

|  |  |
| --- | --- |
| 21 | 49 |
| 22 | 44 |
| 47 | 46 |
| 22 | 12 |
| 26 | 49 |
| 21 | 49 |
| 22 | 44 |
| 47 | 46 |
| 22 | 12 |
| 26 | 49 |

**(五) 執行結果、討論與心得**

執行結果與討論 (充電站站點個數、和生成座標等問題討論)等…

1. 執行結果

Output of program:由於第六題直接是將原先前幾個座標複製一次以達到N = M 的Hungarian method條件，雖然結果會不同，也很難偵錯。

2. 討論

充電站站點個數、和生成座標等問題討論

1. 充電站站點個數K

i.每一站固定讓一台車使用，則為第五題的Hungarian method，座標的數量需為define在標頭檔的#define POINTS\_COUNT 10，N個充電站和M個電動車子的K都為1。

ii.每一站讓k台車使用，第六題的題目，採取直接把同一個座標複製一次。

(2)生成座標

i. K = 1 於createStation的檔案中define的K預設為1

ii.若K != 1則createStation中define的另一個數 M (預設為10) 則要改成 POINTS\_COUNT/K 的答案 (於另一個檔案中define 預設10)。

要注意POINTS\_COUNT的定義要與M\*K 和 N 一樣

3. 心得

如果回傳更正過時間複雜度空間複雜度和演算法則+理論分析之原報告應該對我理解能幫助不少，最好程式碼也更正一下，不然答案向同學對實在很不確定，有時對方說答案不一樣結果可能是兩位都錯，好一點一正一負，不過要每位同學更正不太有可行性，人太多了。

這次作業後兩題的複雜程度真的比較高，不過也有前面幾題基本題，符合人道，並練習了很多操作，期待公布每題正確的演算法和時間空間複雜度來參考，或是舉個例子程式怎麼寫會更好或有更能改善的條件。

可笑是我不知道exe打開來馬上就關掉的原因，於是沿用上次作業的方法，輸入數字以結束程式 return 0。

**參考文獻**

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms," Third Edition, The MIT Press, 2009.
2. 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.
3. Anany V. Levitin, "Introduction to the Design and Analysis of Algorithms," 2nd Edition, Addison Wesley, 2007.
4. Richard Neapolitan and Kumarss Naimipour, "Foundations of Algorithms," Fourth Edition, Jones and Bartlett Publishers, 2010.
5. <https://ccjou.wordpress.com/2012/05/09/%E5%88%A9%E7%94%A8%E8%A1%8C%E5%88%97%E5%BC%8F%E8%A8%88%E7%AE%97%E5%A4%9A%E9%82%8A%E5%BD%A2%E9%9D%A2%E7%A9%8D/>
6. <http://www.csl.mtu.edu/cs4321/www/Lectures/Lecture%206%20-%20Brute%20Force%20Closest%20Pair%20and%20Convex%20and%20Exhausive%20Search.htm>