

# Homework 02

## Question 1

for MEG304301

張皓鈞 B11030202

題目描述: HW2. N66304

Q1: Base

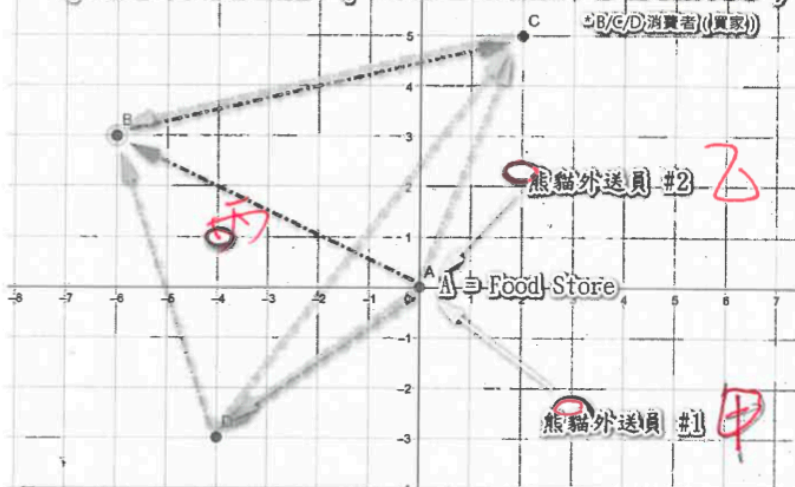
[3對3: 路徑選擇] 送貨路徑選擇

- I. 3買家的(x,y)座標由你決定, 手動輸入;  
(輸入) 3外送員的座標值, 由亂數產生,  $100 < (x, y) < 1000$ .  
請參考以下二次說明和圖例, 列印如下:

輸出  
格式

- 外送員的直線距離: (1) , (2) , (3) , 由 (4) 得標準, 免費取餐, 送餐給 (5) (6) 二位買家.
- 得路徑: (1)  (2)  (3)  (4)  (5)  (6)  AC-BC 距離:
- 建議第 (7) 路徑走法.

Q: 曼哈頓距離 (Manhattan Distance)



- 外送員須自行評估, 先到A店家取餐, 探歐氏(直線)距離, 估計是否合適。
- 採(順道送貨到二位買家)策略: 估算(A店取餐, 送到第一位買主P, 再順道到第二位買主Q)之曼哈頓距離。即(AP和PQ二段路綫)。

註: 紫色路徑是AB+BC, 也相當於估算A店取餐, 送貨到(B+C)兩買主的路徑

曼哈頓距離  
(log):

說明: AB段是[水平6+垂直3], BC段是[水平8+垂直2]

AC段是[2+5垂直] CB段是[水平8+2] 7+10=17  
⇒取最小途徑 II (AC+CB) 順時針方向

說明: 在(B,C,D)三買家, 擇其二, 故有(BC), (BD), (CD)三種  
雙貨組合, 配合以上擇一的判斷, 相當於3x2路組合, 擇量.

#1

# Code

View in Jupyter Notebook

## Tests & Results

### Test 1

#### 隨機產生的外送員資訊

第 1 近的外送員為 丙外送員，座標為 (22, -292)，距離為 292.8275943281302

第 2 近的外送員為 乙外送員，座標為 (-800, -255)，距離為 839.6576683387106

第 3 近的外送員為 甲外送員，座標為 (537, -865)，距離為 1018.132604330104

#### 輸入的顧客資訊

顧客B(8, 7)

顧客C(1, 2)

顧客D(3, 4)

### 運行結果

```
11 nearest_customer = find_nearest_customer(customer_pos, [x, y],
```

```
12 )
```

[358] ✓ 0.2s

... 外送最佳方案為

由 丙外送員(22, -292)，至店鋪取餐，然後先送餐給 顧客C(1, 2)，再送餐給 顧客D(3, 4)

### Test 2

#### 隨機產生的外送員資訊

第 1 近的外送員為 丙外送員，座標為 (393, -283)，距離為 484.2912346925143

第 2 近的外送員為 乙外送員，座標為 (-279, -692)，距離為 746.1266648498765

第 3 近的外送員為 甲外送員，座標為 (-665, 482)，距離為 821.3093205364225

#### 輸入的顧客資訊

顧客B(654, 321)

顧客C(987, 654)

顧客D(98, 765)

## 運行結果

```
12 ) )
[397] ✓ 0.2s
... 外送最佳方案為
    由 丙外送員(393, -283)，至店鋪取餐，然後先送餐給 顧客B(654, 321)，再送餐給 顧客C(987, 654)
```

## Test 3

### 隨機產生的外送員資訊

第 1 近的外送員為 丙外送員，座標為 (393, -283)，距離為 484.2912346925143  
第 2 近的外送員為 乙外送員，座標為 (-279, -692)，距離為 746.1266648498765  
第 3 近的外送員為 甲外送員，座標為 (-665, 482)，距離為 821.3093205364225

### 輸入的顧客資訊

顧客B(-456, -789)  
顧客C(-123, 890)  
顧客D(-345, 678)

## 運行結果

嘗試輸入錯誤的顧客座標 1 ?

```
25 inputCustomersPos (
[433] ✓ 7.3s
... 輸入格式錯誤，請重新輸入
```

```
12 ) )
[423] ✓ 0.2s
... 外送最佳方案為
    由 甲外送員(682, 458)，至店鋪取餐，然後先送餐給 顧客D(-345, 678)，再送餐給 顧客C(-123, 890)
```

## 解題心得

該題目為在業界相當常見的應用情景，解該題目並不需要運動到大量的Python技巧，但是需要一些花時間思考該程式的架構，以及計算方式，做該題目可以扎實訓練Python程式語言基礎與一些簡易運算的基礎。