

CPP1702 Coding Assignment 3

設施選址問題

電子檔 2018/04/16 09:00 前上傳至 moodle
書面檔 2018/04/16 09:15 前於課堂繳交

運用在工業與資訊管理的範圍

基本邏輯推導、設施選址。

相關 C++ 課程內容

讀檔、結構、陣列、動態記憶體的運用與配置。

問題背景與概述

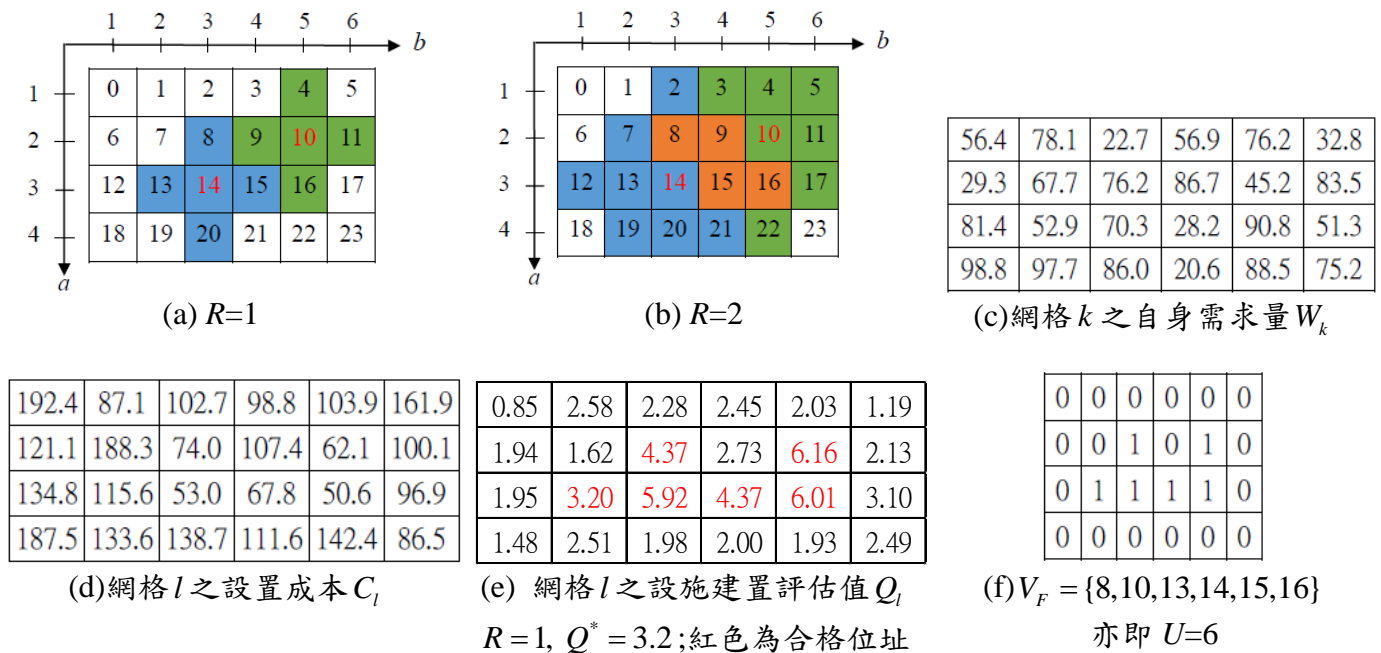
「設施選址」(Facility Location) 是商家針對新設施地點選定的過程，譬如若要開一家新的 7-11 門市，該「設施」應該要設置在哪裡較合宜，就是一個「設施選址」問題。常見的設施選址問題通常可依其候選點位址是否已知而分成兩類：(1)候選點位址未知：此類問題著重在地圖上找到合宜的位址以設置候選點；(2)候選點位址已知：此類問題假設已有數個候選點，欲自其中選出相對合宜的數個位址以設置新設施。本次作業擬讓同學們實作簡化過後的上述兩類議題。

首先，我們將某地區之平面地圖劃分成橫向 M 格、縱向 N 格的 $M \times N$ 個網格區域，而每個網格區域以其中心點座標 (i, j) 代表之，其中 $i=1, \dots, M$ ， $j=1, \dots, N$ 。假設我們以由上而下、由左至右的方式從 0 依序將這些網格編號，亦即第 (i, j) 格之編號為 $(i-1)N + j - 1$ 。為方便起見，我們將這 MN 個網格點所構成之集合稱為 V ，其中編號為 $k=0, 1, \dots, MN-1$ 之網格點座標為 $(V[k].a, V[k].b)$ ，由上述可知 $V[k].b = k \% N + 1$ 而 $V[k].a = (k - V[k].b + 1) / N + 1$ 。假設第 k 網格區域內的顧客總需求量为 W_k ，若在該中心點建置新設施則需花費 C_k ，一個網格區域頂多僅能建置一個設施。

在某網格設置一新設施通常可以服務該網格及其附近的顧客，為方便量測網格間的距離，在此我們定義網格 k 與 l 的間距 d_{kl} 以曼哈頓距離 (Manhattan Distance) $|V[k].a - V[l].a| + |V[k].b - V[l].b|$ 來量測。若某一設施被設置在網格 l ，我們假設與其相距 R 範圍內 (亦即滿足 $d_{lk} \leq R$) 之所有網格 k 其需求加總 $\bar{W}_l = \sum_{k: d_{lk} \leq R} W_k$ 皆可被該設施所服務；倘若與該網格 k 相距 R 範圍內有多個設施，則將僅擇其一來當該網格 k 之專屬負責設施。

設施選址問題大致分成兩階段：**第一階段**先在地圖上找出值得設置的候選位址群，在此我們定義某網格 l 之「設施建置評估值」 $Q_l = \bar{W}_l / C_l$ ，並使用給定的設施建置評估值 Q^* 篩選出合格的候選位址群 $V_F = \{l: Q_l \geq Q^*\}$ ，假設其數量為 $U = |V_F|$ ；**第二階段**再由第一階段選出的候選位址群 V_F 中，挑選出 L 個能滿足較多需求之位址來建置新設施。因第二階段困難度較高，我們設計一種「貪婪演算法」(Greedy Algorithm)來處理，此方法每回合將自當下仍尚未建置新設施的候選網格群中，挑選其可服務之鄰近總需求最大的網格 l_{\max} ，亦即 $\bar{W}_{l_{\max}} = \max_{l \in V_F} \{\bar{W}_l\}$ 。此處的 $\bar{W}_l = \sum_{k: d_{lk} \leq R, B_k=0} W_k$ ，其中布林變數 B_k 記錄網格 l 之某鄰近網格 k 是否在當下已有其專屬之負責設施，若有 (即 $B_k=1$)，則該網格 k 的需求量就不會被納入 \bar{W}_l 以避免重覆計算；反之 (即 $B_k=0$)，則該網格 k 的需求量可被納入 \bar{W}_l 。挑出網格 l_{\max} 之後，更新 $V_F = V_F - \{l_{\max}\}$ 再將此集合 $\{k: d_{l_{\max}k} \leq R, B_k=0\}$ 內之網格 k 設定為 $B_k=1$ 。之後，再重複以上步驟逐一選取最適宜的網格來建置新設施，直至最好的 L 個候選網格被挑完為止。

圖一使用 $M=4, N=6$ 之範例展示相關設定：共有 24 個網格（編號 0 至 23 註明於各格內），網格 10、14 之座標分別為 $(2,5)$ 、 $(3,3)$ ，其中 $5=10\%6+1$ 、 $2=(10-5+1)/6+1$ ； $3=14\%6+1$ 、 $3=(14-3+1)/6+1$ 。如圖一所示，若於網格 10、14 分別設置新設施，隨著服務範圍 R 之數值變大，該設施可服務的範圍亦會變大，甚至造成多重覆蓋。（圖一(b)中之橘色網格）。倘若圖一(c)(d)分別為各網格之自身需求量及設置成本，則圖一(e)為 $R=1$ 設定下之各網格設施建置評估值 Q_l ，舉例來說， $\bar{W}_0=(56.4+78.1+29.3)/192.4=0.85$ ， $\bar{W}_{16}=(90.8+45.2+51.3+28.2+88.5)/50.6=6.01$ ；若 $Q^*=3.2$ 則只有編號為 8,10,13,14,15,16 等 6 個網格夠格可被設置新設施，因此我們將這些合格之網格編號存入一個陣列 $VF[U]$ ，其中 $U=6$ ，而 $VF[0]=8$, $VF[1]=10$, ..., $VF[5]=16$ 。



圖一. $M=4, N=6$ 之範例

假設 $L=3$ ，第二階段將自 $U=6$ 個候選位址 $V_F = \{8, 10, 13, 14, 15, 16\}$ 中，以當下尚未建置新設施且能滿足較多需求之位址為挑選原則，依序挑選出 $L=3$ 個位址以設置新設施。為方便記錄某候選點網格 $VF[l]$ 與某網格 k 之配對或專屬關係，我們建議使用一個 $U \times MN$ 長度的二維布林陣列 $E[U][MN]$ 來記錄，一開始先檢驗網格編號為 $VF[l]$ 之候選點 $l=0, \dots, U-1$ 與網格 $k=0, \dots, MN-1$ 是否相距在 R 範圍內，若是，則設定 $E[l][k]=1$ ，反之 $E[l][k]=0$ 。此陣列在之後將用來記錄各新建置的設施 l 及其所負責的網格 k 之間的專屬關係，針對 $B_k=1$ 之網格 k ，僅存在一個專屬的設施點 l 。舉例來說，倘若與網格 k 相距在 R 範圍內有兩個候選點位於網格 $VF[l_1]$ 、 $VF[l_2]$ ，則一開始時 $E[l_1][k]=E[l_2][k]=1$ 。然而若演算法先選出 l_2 ，再選出 l_1 ，則 $E[l_2][k]=1$ 但 $E[l_1][k]=0$ ，此乃因為該網格 k 已先被指派給網格 $VF[l_2]$ ，為避免重覆計算，就不讓後來才選出的網格 $VF[l_1]$ 服務了。以下說明貪婪演算法步驟：（以下請同時參考第 4 頁定義的 struct 結構與相關變數）

Step 1: 自 U 個候選點 $l=0, \dots, U-1$ 中，只針對目前尚未被設置新設施（即滿足 $V[VF[l]].isF=0$ 之 l ）的候選點，挑選其中 $V[VF[l]].Wbar$ 最大者，代表若於該候選點設置新設施，將滿足最多鄰近網格之需求。假設此步驟挑出第 l_{max} 個候選點，其原始之網格編號則為 $VF[l_{max}]$

Step 2：決定挑選網格 $VF[l_{\max}]$ 來設置新設施，因此設定 $V[VF[l_{\max}]].isF = 1$ ， $L = L - 1$ ，同時將目前與網格 $VF[l_{\max}]$ 相距 R 單位內且尚未被服務的網格 k 指派給網格 $VF[l_{\max}]$ 服務，並解除該網格 k 與其它同樣在 R 距離內的鄰近候選點未來可能的配對關係。

亦即針對 $E[l_{\max}][k] = 1$ 之所有可能的網格 k 設定 $V[k].isB = 1$ ，且針對網格 k 其它同樣滿足 $E[l'][k] = 1$ 的鄰近候選點 l' ，將其原先可能的配對關係重新設定為 $E[l'][k] = 0$ ，以確認該網格 k 之專屬負責設施為網格 $VF[l_{\max}]$ 。

Step 3：重複 Step 1、2，直至 $L = 0$ 為止

以圖一為例，因 $U=6$ 個候選位址 $V_F = \{8, 10, 13, 14, 15, 16\}$ 皆尚未被設置新設施，其個別可能服務的鄰近總需求量 $\bar{W}_l = \sum_{k: d_{lk} \leq R} W_k$ 分別為 323.6, 382.4, 370, 313.6, 296.6, 304，挑選其中最大者 382.4（即候選點 $l_{\max} = 1$ ，網格編號為 $VF[1] = 10$ ），亦即網格 10 來設置第一個新設施，設定 $V[10].isF = 1$ ， $L = 3 - 1 = 2$ ，將其鄰近 1 單位內且尚未被服務的網格 4, 9, 10, 11, 16 皆指派給網格 10 服務，此即設定 $V[k].isB = 1$ ，其中 $k = 4, 9, 10, 11, 16$ ；接著，因為原先的 $E[0][9] = E[4][9] = 1$ 代表網格 9 原先亦可能被指派給位於網格 8、15 之候選點 0、4 服務，然而因為當下我們先選了位於網格 10 之候選點 1 來建置新設施，因此必須重設 $E[0][9] = E[4][9] = 0$ 。同理，重設 $E[5][10] = 0$ ， $E[4][16] = E[5][16] = 0$ 。

再來，我們自 5 個候選位址 $V_F = \{8, 10, 13, 14, 15, 16\}$ 中計算其個別可能服務的鄰近總需求量 $\bar{W}_l = \sum_{k: d_{lk} \leq R} W_k$ 分別為 236.9, 370, 313.6, 119.1, 168，挑選其中最大者 370（即候選點 $l_{\max} = 2$ ，網格編號為 $VF[2] = 13$ ），亦即網格 13 來設置第二個新設施，設定 $V[13].isF = 1$ ， $L = 2 - 1 = 1$ ，將其鄰近 1 單位內且尚未被服務的網格 7, 12, 13, 14, 19 皆指派給網格 13 服務，此即設定 $V[k].isB = 1$ ，其中 $k = 7, 12, 13, 14, 19$ ；接著，因為原先的 $E[0][7] = 1$ 代表網格 7 原先亦可能被指派給位於網格 8 之候選點 0 服務，然而因為當下我們先選了位於網格 13 之候選點 2 來建置新設施，因此必須重設 $E[0][7] = 0$ 。同理，重設 $E[3][13] = 0$ ， $E[0][14] = E[3][14] = E[4][14] = 0$ 。

最後，我們自 4 個候選位址 $V_F = \{8, 10, 13, 14, 15, 16\}$ 中計算其個別可能服務的鄰近總需求量 $\bar{W}_l = \sum_{k: d_{lk} \leq R} W_k$ 分別為 98.9, 190.4, 48.8, 168，挑選其中最大者 190.4（即候選點 $l_{\max} = 3$ ，網格編號為 $VF[3] = 14$ ），亦即網格 14 來設置第三個新設施，設定 $V[14].isF = 1$ ， $L = 1 - 1 = 0$ ，將其鄰近 1 單位內且尚未被服務的網格 8, 15, 20 皆指派給網格 14 服務，此即設定 $V[k].isB = 1$ ，其中 $k = 8, 15, 20$ ；接著，因為原先的 $E[0][8] = 1$ 代表網格 8 原先亦可能被指派給位於網格 8 之候選點 0 服務，然而因為當下我們先選了位於網格 14 之候選點 3 來建置新設施，因此必須重設 $E[0][8] = 0$ 。同理，重設 $E[4][15] = E[5][15] = 0$ 。綜上所述，上述例子逐一挑選了網格 10、13、14 來建置新設施，總共滿足了 $382.3 + 370 + 190.3 = 942.6$ 需求量，服務了編號 4, 9, 10, 11, 16; 7, 12, 13, 14, 19; 15, 20 等 13 個網格。

作業目的及假設

本次作業主要分三部分：PART1 讀檔、以 struct 陣列建構各網格 l 之相關資料（包含座標、自身需求量、建置成本等），此部分主要讓同學練習動態設置陣列以及 struct；PART2 進一步計算各網格 l 可涵蓋之總需求量 $\bar{W}_l = \sum_{k: d_{lk} \leq R} W_k$ ，並篩選出合格的候選位址群 $V_F = \{l: Q_l \geq Q^*\}$ ，此部分訓練同學理解與邏輯推理能力；PART3 實作貪婪演算法，自 V_F 中以貪婪法則選取 L 個能滿足較多需求之位址來建置新設施，此部分的邏輯想法較複雜，亦訓練同學理解與邏輯推理能力。

本題假設每網格頂多只能設置一個設施，PART3 著重於理解演算法之設計緣由，而非求取最佳解。

程式要求及作法

本作業大概可切割成以下三個 PART：(單一程式)

PART1[20%] 宣告與讀檔：

請使用相同的命名方法宣告下列結構、變數及動態陣列，額外需求的自訂變數可自行命名。

```
struct Zone{           //此為各網格之 struct
    int a;              //儲存某網格之縱軸座標，即由上而下第 a 列， a=1,...,M
    int b;              //儲存某網格之橫軸座標，即由左而右第 b 行， b=1,...,N
    double W,           //儲存某網格之自身需求量
           C,           //儲存某網格之設施建置成本
           Wbar,        //儲存某網格可涵蓋之鄰近 R 範圍內之需求量總和
           Qi;          //儲存某網格之設施建置評估值 = Wbar/C
    bool isQ,           //儲存某網格是否合格得以建置新設施，合格則為 1，否則為 0
         isF,           //儲存某網格是否被挑選來建置新設施，是則為 1，否則為 0
         isB;           //儲存某網格是否已被指派其專屬之設施，是則為 1，否則為 0
};

int M,                 //區域縱軸座標範圍(1,2,...,M)
    N,                 //區域橫軸座標範圍(1,2,...,N)
    U,                 //合格之候選網格個數，計算完各網格之設施建置評估值才可得到
    L,                 //欲選擇設立之設施個數
    R;                 //設施涵蓋距離範圍，使用曼哈頓距離量測

double Qstar;          //給定的設施建置評估值

Zone *V;               //動態宣告 MN 長度之網格 struct 陣列; V[k], k=0,1,...,MN-1
int *VF;               //動態宣告 MN 長度之設施編號陣列; VF[l], l=0,1,...,MN-1
bool **E;              //動態宣告  $U \times MN$  長度的二維布林陣列 E[U][MN]，用於記錄是否
                        //候選點  $l=0, \dots, U-1$  是否與網格  $k=0, \dots, MN-1$  有專屬負責關係
```

請使用者輸入檔名，譬如讀取 [data1.txt](#) 檔案後，其格式如下：

4 6 3.2 3 1	(M N Q* L R)
56.4 78.1 22.7 56.9 76.2 32.8	(需求量 $W_0, W_1, W_2, \dots, W_{N-1}$)
29.3 67.7 76.2 86.7 45.2 83.5	(需求量 $W_N, W_{N+1}, W_{N+2}, \dots, W_{2N-1}$)
81.4 52.9 70.3 28.2 90.8 51.3	(需求量 $W_{2N}, W_{2N+1}, W_{2N+2}, \dots, W_{3N-1}$)
98.8 97.7 86.0 20.6 88.5 75.2	(需求量 $W_{3N}, W_{3N+1}, W_{3N+2}, \dots, W_{4N-1}$)
192.4 87.1 102.7 98.8 103.9 161.9	(建置成本 $C_0, C_1, C_2, \dots, C_{N-1}$)
121.1 188.3 74.0 107.4 62.1 100.1	(建置成本 $C_N, C_{N+1}, C_{N+2}, \dots, C_{2N-1}$)
134.8 115.6 53.0 67.8 50.6 96.9	(建置成本 $C_{2N}, C_{2N+1}, C_{2N+2}, \dots, C_{3N-1}$)
187.5 133.6 138.7 111.6 142.4 86.5	(建置成本 $C_{3N}, C_{3N+1}, C_{3N+2}, \dots, C_{4N-1}$)

輸出以下結果：M,N,Q*,L,R; W[], C[]

Enter filename: data1.txt

M=4; N=6; Q*=3.2; L=3; R=1;

W	1	2	3	4	5	6
1	56.4	78.1	22.7	56.9	76.2	32.8
2	29.3	67.7	76.2	86.7	45.2	83.5
3	81.4	52.9	70.3	28.2	90.8	51.3
4	98.8	97.7	86	20.6	88.5	75.2

C	1	2	3	4	5	6
1	192.4	87.1	102.7	98.8	103.9	161.9
2	121.1	188.3	74	107.4	62.1	100.1
3	134.8	115.6	53	67.8	50.6	96.9
4	187.5	133.6	138.7	111.6	142.4	86.5

PART2[40%] 計算 \bar{W}_i 、 Q_i 、 V_F 、 U 、 E ：

此部分較難的是構思 \bar{W}_i 的計算方式，如何找到相距 R 內的網格，作法不只一種，差別在於效率，老師鼓勵同學們設計更有效率的作法（若足夠特別，可額外多加 10 分）。之後的 Q_i 、 V_F 、 U 都相對容易處理。需要注意的是 VF 陣列的長度 U ，理應在算出 Q_i 之後即可得知，並可據以動態宣告設置 VF 陣列，在此我們不特別規定一定要如此做，亦即你可以先將 VF 陣列的長度宣告長度為 MN 但其實只用到前 U 個。另外，為方便對答案，本作業要求存於 VF 陣列的網格編號應由小而大儲存。

輸出以下結果：U,VF[], Wbar[], Q[], E[][][] （除 Q[] 顯示至小數點 2 位，其餘皆 1 位即可）

U=6, VF: 8 10 13 14 15 16

w_bar	1	2	3	4	5	6
1	163.8	224.9	233.9	242.5	211.1	192.5
2	234.8	304.2	323.6	293.2	382.4	212.8
3	262.4	370.0	313.6	296.6	304.0	300.8
4	277.9	335.4	274.6	223.3	275.1	215.0

Q	1	2	3	4	5	6
1	0.85	2.58	2.28	2.45	2.03	1.19
2	1.94	1.62	4.37	2.73	6.16	2.13
3	1.95	3.20	5.92	4.37	6.01	3.10
4	1.48	2.51	1.98	2.00	1.93	2.49

E	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	0

PART3[40%] 實作貪婪演算法，逐一算出 L 個挑選出來的候選網格：

此部分必須先看懂題目描述的演算法步驟，較難的是在選取當下最好的候選網格後，如何更新相關的網格與設施之專屬關係，以便計算更新過後的 \bar{W}_i 。其實， \bar{W}_i 的計算方式也有很多種，同學們可自行發揮。此部分應輸出各回合（總共 L 回合）所挑選出的最佳候選網格，以及該次挑選所新服務的新網格編號及其總需求量。以 data1.txt 為例，理應輸出：

Iteration 1: select Facility 10, satisfied 382.4 demands for zone 4,9,10,11,16,
 Iteration 2: select Facility 13, satisfied 370.0 demands for zone 7,12,13,14,19,
 Iteration 3: select Facility 14, satisfied 190.4 demands for zone 8,15,20,
 Total satisfied demands: 942.8

解題建議

- 老師估計初學者至少花 5hr 才能將本題作好，因為下週放假，且 4/9 舉行期中考，老師特地延後繳交日期至 4/16。然因本次作業之部分內容可能跟 4/9 考試相關，所以仍建議同學能早點做完。如果你在 HW2 必須花超過 5hr 才能作好，很可能這題會花你 9hr 以上。或許你可能仍對題目中常出現的數學符號或式子有排斥或恐懼感，但其實這些都是本系專業中非常基本常見的符號表示方式，所以還是請自行調整心態。本次作業的敘述，每字句都是琢磨過的，例子數值的呈現也都是設計過的，你若有哪裡看不懂，或不清楚哪個數字是怎麼算出來的，請再仔細多研讀幾次，試著推敲理解，不要太快就攤手投降。建議你可用紙筆方式依題目論述的流程實地去推導演練，如此應可幫自己更快弄清楚來龍去脈。
- 雖說本題繳交日期延至 4/16，但 HW4 極可能在 4/10 公布，且其難度又較 HW3 高，所以還是建議同學們真的要儘早開始寫本次作業、儘早將它結束掉。
- 本題分成 3 個 PART 處理，除 PART1 應該可以很快解決（.5hr 內），PART2 跟 PART3 應皆需花一些時間處理。本次作業應依 PART1、2、3 的順序才能完成，建議命名為 **hw3.cpp**
- 本題的 4×6 例子其實也可以自行用 EXCEL 演練看看。
- 除 [data1.txt](#) 及 [其結果](#) 外，請自行測試 [data2.txt](#) 及檢驗 [其結果](#)。

*作業繳交應注意事項

1. 作業需繳交電子檔以及書面(列印程式檔及程式結果)
2. 電子檔請於作業繳交截止時間以前上傳至 <http://moodle.ncku.edu.tw>
 - 2.1 請同學先建立一個資料夾，資料夾名稱為“學號_hw3”，例如學號為 hxxxx，則資料夾名稱則為 **hxxxx_hw3**
 - 2.2 將程式檔案名稱存為“hw3.cpp”，並將此程式檔案存於上述設立之學號_hw3 資料夾中
 - 2.3 最後將整個學號_hw3 資料夾壓縮成 zip 檔(學號_hw3.zip)，再上傳至 moodle 系統

(！注意！：[請勿](#)將 cpp 檔 copy/paste 至 word 檔而上傳之)
3. 書面作業請於 **2018/04/16 上課 5 分鐘內(09:15 前)**繳交至講台，其中需要註明程式是否能被編譯與執行、撰寫人、程式之目的、如何編譯及執行等資料(詳見 <https://www.dropbox.com/s/pvac59tfefggokd/programming.html?dl=0>)。