

國立高雄科技大學

離散數學 期末報告

班級：電子工程系三甲（第一）

姓名：李步剛

學號：C111112102

任課老師：梁毓珊

期末報告說明

- (1) 請寫成 Word 檔
- (2) 自期中考後所學的每個單元，各寫一頁延伸學習內容，各單元如下
- (3) 所寫內容不可重複課堂上學過的演算法或範例

07-演算法

找一個你覺得最有趣的演算法，說明為什麼有趣，說明內容，並將其流程寫成演算法的格式

08-計數的基礎

設計一個例子，同時應用乘法法則與加法法則

09-鴿洞原理

設計一個例子，應用鴿洞原理的延伸

10-排列與組合

設計四個例子，分別是：依序選取可複選、依序選取不可複選、無選取順序不可複選、無選取順序可複選

11-圖

舉一個應用圖論來解決的問題，介紹該問題及解法
最後

本學期的學習心得&課堂建議

演算法:天氣預測演算法-數值天氣預報

為何覺得有趣:

從小的時候就對天氣感興趣，而且對於新聞台的天氣預報覺得不可思議，想說人類如何得知天氣未來走向。國高中的地科課所學的氣象知識讓我更了解天氣變化的種種原因，但始終對天氣預測存在著不解，想說人類怎能如此厲害，天氣都有辦法預測。今日離散數學課程中的演算法單元，讓我得知原來天氣預測是跟演算法有關，正好讓我這個對於天氣有興趣且對預測有疑惑的人可以好好深入了解。

說明:

天氣預測演算法通過分析氣象站、衛星及其他來源的龐大數據，預測未來的天氣變化，為人們的出行和活動提供指導。全球預報系統（GFS）和歐洲中期天氣預報中心（ECMWF）等系統就是基於此類技術運行的高精度預測模型-數值天氣預報。

數值天氣預報是最傳統且廣泛使用的技術，基於物理學搭配數學方程式的模擬來預測天氣。這些數學方程式可以簡化以概略描述大氣系統，或可更複雜化以精確地模擬大氣行為。然而，隨著數學系統變得複雜，我們無法以解析方式求解，只能使用數值方法來接近真實解，透過數值模式模擬大氣運動。隨著大氣科學、數學與數值方法的進步，數值模式更加精確，使天氣預報也變得更準確。

運作方法:

使用大氣動力學和熱力學的方程式（如納維-斯托克斯方程式），模擬大氣中的流體運動。

所需要的數據:

1. 氣溫、濕度、氣壓、風速等地面和高空觀測數據。
2. 衛星圖像、雷達觀測資料。

常用模型:

1. ECMWF（歐洲中期天氣預報中心）
2. GFS（全球預測系統）

根據演算法之模型，我們需定義輸入及輸出

● **輸入：**

觀測數據：

地面氣象站（溫度、氣壓、濕度、風速等）

衛星數據（雲層影像、輻射值）

雷達數據（降水分布、風速切變）

歷史氣象資料

初始條件：

初始大氣狀態（氣壓、溫度、風場）

模型網格（空間、單位時間）

邊界條件：

水平邊界條件：計算區域的水平方向邊緣（東、西、南、北邊界）

垂直邊界條件：計算區域的垂直方向邊緣（對流層頂和地表）

● **輸出：**

時間序列的氣象參數：

1. 氣溫、氣壓、風速、濕度、降水量等
2. 空間分布圖（如等壓線圖、風場圖）
3. 天氣趨勢分析（未來幾小時到幾天的預測）

演算法實作：

模式運作解釋（為降低複雜度，以下以溫度作為舉例）：

初始資訊：

1. 模型大小長 X 和寬 Y 的大小設定為 10×10
2. 初始溫度場除邊界外全部設為 25°C

邊界條件（為降低演算法知複雜度，因此只考慮水平邊界）：

四個邊界的溫度設為 30°C ，邊界之溫度會由高至低影響內部點之溫度

更新溫度場：

使用鄰域平均更新內部點的溫度值(每個內部點的新溫度是它四個鄰近點(上下左右)的平均值)

時間點：

模擬之 time_step(時間單位)，每 1 個時間單位更新一次溫度場並顯示整個模型各個點之溫度)，每 5 個時間單位顯示一次中心點的溫度

根據以上運作模式，得出輸入(模型大小長 X 與寬 Y、執行之 time_step(時間單位))與輸出(中心點溫度)、整個溫度場之溫度。

模型示意圖(10 x 10 範圍，最外圈邊界為 30°C，內部均為 25°C)：

30 索引 0	30 索引 1	30 索引 2	30 索引 3	30 索引 4	30 索引 5	30 索引 6	30 索引 7	30 索引 8	30 索引 9
30 索引 10	25 索引 11	25 索引 12	25 索引 13	25 索引 14	25 索引 15	25 索引 16	25 索引 17	25 索引 18	30 索引 19
30 索引 20	25 索引 21	25 索引 22	25 索引 23	25 索引 24	25 索引 25	25 索引 26	25 索引 27	25 索引 28	30 索引 29
30 索引 30	25 索引 31	25 索引 32	25 索引 33	25 索引 34	25 索引 35	25 索引 36	25 索引 37	25 索引 38	30 索引 39
30 索引 40	25 索引 41	25 索引 42	25 索引 43	25 索引 44	25 索引 45	25 索引 46	25 索引 47	25 索引 48	30 索引 49
30 索引 50	25 索引 51	25 索引 52	25 索引 53	25 索引 54	25 索引 55	25 索引 56	25 索引 57	25 索引 58	30 索引 59
30 索引 60	25 索引 61	25 索引 62	25 索引 63	25 索引 64	25 索引 65	25 索引 66	25 索引 67	25 索引 68	30 索引 69
30 索引 70	25 索引 71	25 索引 72	25 索引 73	25 索引 74	25 索引 75	25 索引 76	25 索引 77	25 索引 78	30 索引 79
30 索引 80	25 索引 81	25 索引 82	25 索引 83	25 索引 84	25 索引 85	25 索引 86	25 索引 87	25 索引 88	30 索引 89
30 索引 90	30 索引 91	30 索引 92	30 索引 93	30 索引 94	30 索引 95	30 索引 96	30 索引 97	30 索引 98	30 索引 99

根據輸入及輸出並搭配運作方式及模型以程式碼(C 語言)呈現:

(注意:以下使用之陣列第一位為索引 0)

```
#include <stdio.h>

#define X 10 // 模型(溫度場)長度大小 (X 方向) , 設為 10
#define Y 10 // 模型(溫度場)寬度大小 (Y 方向) , 設為 10
#define time_step 20 // 時間單位設為 20
void update_temperature(double temp[], int x, int y) ; // 子函式宣告 , 更新溫度場中之溫度

int main() {
    int model_size = X * Y ; //模型(溫度場)大小
    double temperature[model_size] ; // 一維浮點陣列表示模型(溫度場)

    // 初始化溫度場 , 假設模型(溫度場)內每一點初始溫度均為 25 度
    for (int i = 0; i < model_size; i++) {
        temperature[i] = 25 ;
    }

    // 設定邊界條件
    for (int i = 0; i < X; i++) {
        temperature[i] = 30 ; // 上邊界(索引 0 至 9)
        temperature[ ( ( Y - 1 ) * X ) + i ] = 30 ; // 下邊界(索引 90 至 99)
    }
    for (int j = 0; j < Y; j++) {
        temperature[ j * X ] = 30 ; // 左邊界(索引 0、10、20、30.....90)
        temperature[ ( j * X ) + X - 1 ] = 30 ; // 右邊界(索引 9、19、29、39.....99)
    }

    // 模擬過程
    for (int step = 0; step < time_step; step++) {
        update_temperature(temperature, X, Y) ; // 更新溫度場中之溫度
        printf("單位時間 %d 時 , 溫度場中各點溫度為 :\n", step) ;

        //顯示溫度場中各點之溫度
        for (int i = 0; i < model_size; i++) {
            printf("%f ", update_temperature[i]) ;
```

```

        if ( ( i+1 ) % X == 0 ) // 每輸出 X 個資料時即換行(此範例 X 為 10)
            printf("\n");
    }

    // 每 5 個時間單位顯示一次中心溫度
    if (step % 5 == 0) {
        int center_index = ( (Y / 2) * X ) + ( X / 2 ); //溫度場之中心點
        printf("中心溫度為 %f \n", update_temperature[center_index]);
    }
}
return 0;
}

// 子函式，更新溫度場之溫度
void update_temperature(double temp[], int x, int y) {
    double new_temp[x * y]; // 用於存儲更新後的溫度

    //鄰域平均方法更新內部點的溫度(只計算邊界以內之點)，以下連續計算兩點
    for (int j = 1; j < y - 1; j++) {
        for (int i = 1; i < x - 1; i++) {
            int index = j * x + i;
            new_temp[index] = (temp[index - 1] + // 左
                               temp[index + 1] + // 右
                               temp[index - x] + // 上
                               temp[index + x]) / 4; // 下
        }
    }
}

```

點一：將 $x=10$, $y=10$ 代入，假設要改變除邊界外之第一點(索引 11)的溫度($i=1$, $j=1$ 時)，index 為 11 代入計算。

取索引 11 之左點 10(左邊界，溫度為 30)；取其之右點 12(溫度為 25)；取其之上點 1(上邊界，溫度為 30)；取其之下點 21(溫度為 25)，四者相加後平均，得出索引 11 之新溫度 **27.5**。

點二：將 $x=10$, $y=10$ 代入，假設要改變除邊界外之第二點(索引 12)的溫度($i=2$, $j=1$ 時)，index 為 12 代入計算。

取索引 12 之左點 11(更新點，溫度為 **27.5**)；取其之右點 13(溫度為 25)；取其之上點 2(上邊界，溫度為 30)；取其之下點 22(溫度為 25)，四者相加後平均，得出索引 12 之新溫度 **26.875**。

以此類推，一一計算完邊界內所有點之溫度。

// 將計算完成之新溫度更新至原始資料陣列，經由主函式列印出結果。

```
for (int j = 1; j < y - 1; j++) {  
    for (int i = 1; i < x - 1; i++) {  
        int index = ( j * x ) + I ;  
        temp[index] = new_temp[index];  
    }  
}  
}
```


計數的基礎

說明：

地球是孕育生命的源頭，現今地球上繁多的生命種類都是因為地球上自然現象，也造就我們所謂的天氣，同時也有晝夜變化，成為周遭環境。組成天氣的元素可簡單分成溫度、濕度、風速……等等，加上晝夜變化，其中個別元素又包含以下情形(以下僅大致分級，實際上分級更複雜):

1. 溫度(以當天溫度做一個相對比較):低溫、中溫、高溫(3 種)
2. 濕度:潮濕、乾燥(2 種)
3. 風速:無風、微風、和風、強風、疾風(5 種)
4. 雲層:晴天、陰天(2 種)
5. 晝夜:白天、黑夜(2 種)

應用：

乘法法則：

首先，以這些元素組成一個環境，不考慮實際狀況，
即有 $3 \times 2 \times 5 \times 2 \times 2 = 120$ 種環境組成。

但其實一般來說，因為白天太陽輻射加熱地表，地表溫度升高；而到了晚上，太陽輻射停止，地表開始釋放熱量，加上夜晚大氣層的熱輻射無法完全補償地表熱量的散失，因此氣溫下降。

因此黑夜溫度通常白天溫度較低，所以環境組合中，白天不能搭配低溫，黑夜不能搭配高溫，因此分為兩者討論得出以下結果:

白天: $2 \times 2 \times 5 \times 2 \times 1 = 40$ 種環境組成

黑夜: $2 \times 2 \times 5 \times 2 \times 1 = 40$ 種環境組成

加法法則：

以上乘法法則只能分別討論白天與黑夜，接著利用加法法則得出一天中整個環境的種類組成，表示白天與黑夜的環境組成元素不能同時成立:

白天環境組成 + 黑夜環境組成 = $40 + 40 = 80$ 種環境組成。

鴿洞原理

說明：

地球因為自轉的關係，所以存在著一種假想力-科氏力，加上天氣的影響，在暖洋地區會因水蒸氣蒸發猛烈，造成低氣壓出現，最後形成熱帶風暴(北大西洋、北太平洋中部和東北太平洋，被稱為「颶風」，西北太平洋被稱為「颱風」，南太平洋和印度洋則被稱為「旋風」)。

至於熱帶風暴的名稱是由颱風委員會成員國的 14 個國家各取 10 個，總共 140 個名稱來輪流命名。

問題：

根據氣象局資料，2024 一整年在西北太平洋共有 26 個颱風形成，假設 140 個名稱均只用在西北太平洋所形成的颱風且將 26 設為每年平均颱風形成個數，請問在西北太平洋形成相同名稱的颱風需要幾年的時間？

解答：

令原始鴿子數(颱風數量)為 26 隻(個)，一年增加 26 隻(個)， n 年後為 $26n$ 隻(個)

令鴿洞數(名稱數)140 個

套用至廣義的鴿洞原理隻定義，若 $26n$ 隻鴿子住進 140 個鴿洞，則至少有一個鴿洞包含至少 $\left\lceil \frac{26n}{140} \right\rceil$ 個物件。

題目要求出現 2 次相同名稱的颱風，也就是要有一個鴿洞包含兩隻鴿子，即 $\left\lceil \frac{26n}{140} \right\rceil = 2$

得出算式：

$$26n = (140 \times 1) + 1 = 141 \Rightarrow n \text{ 約為 } 5.423$$

表示大約 5.423 年就會出現一個相同名稱的颱風。

排列與組合

依序選取可複選

問題：

根據氣象局的天氣定義，大致可分為以下分級(只約略挑選幾項)

晴:晴天、晴時多雲、多雲時晴

陰:多雲、多雲時陰、陰天

雨:多雲陣雨、雨天、多雲雷陣雨、午後短暫雷陣雨

雪:下雪、積冰、暴風雪

現今氣象局表示某地一天大致之天氣狀況時，均會使用以上之天氣描述作為輔助，協助民眾更快速了解一天之天氣情況。

假設氣象局一次發布一週之天氣預報，採用以上各種天氣描述，請問這一週會有多少天氣組合？

解答：

根據以上天氣描述之種類，晴有 3 種、陰有 3 種、雨有 4 種、雪有 3 種，總和為 13 種。

每天的天氣情況均可從這幾種描述重複挑選，

因此一週 7 天共有 $13^7 = 62748517$ 種天氣組合。

依序選取不可複選

問題：

假設氣象局有天氣預報員專門在向外界預報天氣。假設目前在職預報員共有七位，分別是預報員 1 號、預報員 2 號、預報員 3 號……預報員 7 號，分別安排於一週各天上班。公司內部規定一個星期內一個預報員只能預報一次天氣，加上有兩位預報員(1 號、3 號)因個人原因，1 號不能在星期天上班，3 號只能在星期二上班，請問對於一週的天氣預報，公司共有幾種排班組合？

解答：

整理題目條件：

1. 一共有七位預報員（1 號、2 號、3 號、4 號、5 號、6 號、7 號）
2. 每位預報員在一週內只能值班一次。
3. 每天只能有一位預報員值班。
4. 1 號不能在星期天上班。
5. 3 號只能在星期二上班。

步驟一：

考慮 1、3 的個別情況，需先討論排班彈性最小之人員

3 號只能於星期二上班，因此星期二固定給 3 號，有 1 種排法。

1 號不能於星期天上班，又星期二已分派給 3 號，因此有 5 種排法。

步驟二：

討論其他無限制的人員，剩下五位安排於其他五天，有 $5! = 120$ 種排法。

將所有可能性作相乘 \Rightarrow 共有 $1 \times 5 \times 120 = 600$ 種排班組合。

無選取順序不可複選

問題：

根據氣象局的天氣定義，大致可分為以下分級(只約略挑選幾項)

晴:晴天、晴時多雲、多雲時晴

陰:多雲、多雲時陰、陰天

雨:多雲陣雨、雨天、多雲雷陣雨、午後短暫雷陣雨

雪:下雪、積冰、暴風雪

假設每週氣象局研究員想由氣象預報所呈現的資料進行分析、修正及驗證，以確保對外界呈現的天氣資料的完整性及精確度。以上天氣定義沿用依序選取可複選單元，晴有三種、陰有三種、雨有四種、雪有三種，四種項目總共有 13 種天氣描述。

如果研究員想在這四項中，每天挑選一種進行研究，不可重複挑選(因為一種資料每週只需研究一次)，然而台灣因地理位置的關係，下雪之現象並不常見，因此一週內最多只能從中選取一種來研究，請問一週有幾種選取的可能？

解答：

分別考慮是否挑選雪之描述的選取可能(選取天氣資料不需考慮順序)

情況一：選擇一種雪之描述(最多一種)

雪：從三種描述選取一種 $\Rightarrow C_1^3 = 3$ 種選擇

剩下 6 種從其他非雪之天氣描述(晴、陰、雨)挑選

$$\Rightarrow C_6^{10} = \frac{10!}{6! \times 4!} = 210 \text{ 種選擇}$$

情況一有 $3 \times 210 = 630$ 種選取可能。

情況二：不選擇雪之描述

雪：0 種選擇

7 種均從其他非雪之天氣描述(晴、陰、雨)挑選

$$\Rightarrow C_7^{10} = \frac{10!}{7! \times 3!} = 120 \text{ 種選擇}$$

情況二有 120 種選取可能。

兩種情況總和為 $630 + 120 = 750$ 種選取可能。

無選取順序可複選

問題：

氣象觀測站用於觀測當地各天氣指標，根據演算法單元，列出幾項觀測指標：溫度、氣壓、濕度、風速，透過這些指標可以組成一個天氣的狀況。假設氣象局有一批新進的觀測員，共十位。公司以實際操作的方式訓練新進員工，每人均從四種指標挑選一個來進行觀測，但每項指標必須至少一人選擇，請問這十位觀測員共有幾種選法？

解答：

步驟一：

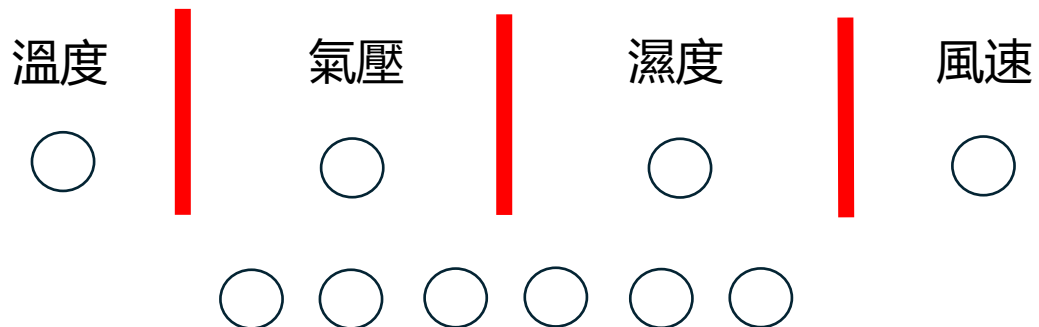
因每項指標必須至少一人選擇，因此將其中四位分配至各指標，因此人數剩餘六位。

步驟二：

剩下六位可自由分配，且每項指標可被重複選擇。根據重複組合單元中概念，四項指標可由三界線作為區隔，加上剩餘的六人，總計可視為九個物品，套用公式

$$\Rightarrow C_6^9 = \frac{9!}{6! \times 3!} = 84 \text{ 種可能 (表示從四個物體中取六個物體，可重複)}$$

(圓圈為人員)





說明：

在天氣相關的應用中，圖論常常被用來解決一些與路徑規劃、資源分配和災害管理有關的問題。在天氣災害發生時，需要根據當時的交通狀況（如道路阻塞、洪水或其他災害造成的影響）來選擇最佳的撤離路徑。每條道路的通行條件會受到災害影響，這些影響可以被視為路段的成本，而我們的目標是找到從災區到避難所的最低成本路徑。這樣的路徑選擇問題可以透過圖論中的最短路徑算法 - **Dijkstra 演算法**(戴克斯特拉演算法)來解決，以確保在災難情況下能夠迅速且有效地運送救援物資與民眾到達安全地區。

戴克斯特拉演算法：

運用前提：

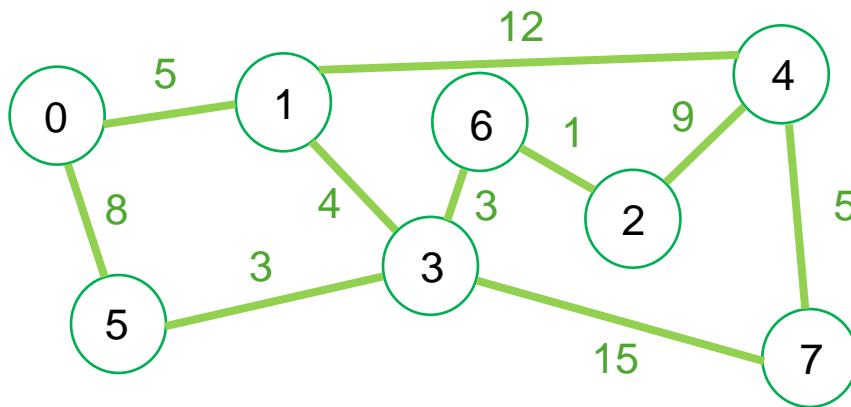
1. 圖的權重：所有邊的權重必須為非負數。
2. 圖的結構：適用於有向圖或無向圖。
3. 單源最短路徑：只能計算從一個起點到其他單節點的最短路徑。
4. 圖的連通性：起點必須能到達其他節點，否則該節點距離為無窮大。
(權重的含義：邊的權重表示距離、時間或花費等非負值)

步驟：

1. 初始化：
創建一個未訪問節點的集合。裡面包含許多節點，並選擇起點。
設定所有節點與起點的初始距離為無窮大(除了起點，其距離為 0)。
2. 選擇當前節點：
選擇未訪問節點中距離最短的節點作為當前節點。
3. 更新鄰近節點的距離：
對於當前節點的每個鄰接節點(不考慮當前節點)，檢查是否通過當前節點的路徑更短。如果更短(小於)，則更新該鄰接節點的距離。
4. 標記當前節點為已訪問：
將當前節點標記為已訪問，並將其從未訪問集合中移除。
5. 重複步驟 2-4：
重複以上步驟，直到所有節點都被訪問過或終點的最短路徑已確定。
6. 結果：
從起點到終點的最短路徑即為最佳路徑，並且可以得到這條路徑的最小「成本」，即時間、距離等。

問題：

(線條長度與實際距離沒關係)



假設因天氣造成災害的地方為節點 0，而避難所位於節點 7，請問最短距離為何及最短路徑為何？

解答：

(以下每更動的之資料使用紅色標示)

1. 初始化

選定起點並標記。設定其他節點與起點距離為無窮大，與自己距離為 0。

節點	與起點距離	前節點
0(起點) ✓	0	無
1	∞	尚無
2	∞	尚無
3	∞	尚無
4	∞	尚無
5	∞	尚無
6	∞	尚無
7	∞	尚無

2. 更新鄰近節點的距離、選擇當前節點、標記當前節點為已訪問

(注意:計算與起點距離時,必須由前一步驟時找出的未標記中與起點最近距離之節點開始,接著由前節點向前推算,於第二步說明)

第一步:

從 0 開始,其相鄰的點有節點 1(與起點距離 5)與節點 5(與起點距離 8),因為 5 與 8 均小於 ∞ ,所以替換其與起點距離並標記其前節點為 0。判斷未標記中,節點 1 為與起點距離最短,將其標記。

節點	與起點距離	前節點
0(起點)	0	無
1 ✓	5	0
2	∞	尚無
3	∞	尚無
4	∞	尚無
5	8	0
6	∞	尚無
7	∞	尚無

第二步:

從上步驟所得出未標記中與起點最近距離之節點(節點 1),其相鄰的點有節點 3(與起點距離 9)與節點 4(與起點距離 17),因為 9 與 17 均小於 ∞ ,所以替換其與起點距離,並標記其前節點為 1。

判斷未標記中,節點 5 為與起點距離最短,將其標記。

(該如何求節點 4 與起點之距離:

1. 從節點 4 往上步驟未標記中與起點最近距離之節點(節點 1)開始推算(不可選擇節點 2 或節點 7),節點 3 與節點 1 之距離為 12。
2. 接著再往節點 1 之前節點(節點 0)推算,節點 1 與節點 0 之距離為 5。
3. 節點 0 為起點,沒有前節點,因此距離推算完畢。將前面所計算的距離作相加 $\Rightarrow 12 + 5 = 17$,得出目前節點 4 與起點之距離。)

其他節點依此類推。

節點	與起點距離	前節點
0(起點)	0	無
1 ✓	5	0
2	∞	尚無
3	9	1
4	17	1
5 ✓	8	0
6	∞	尚無
7	∞	尚無

第三步：

從上步驟所得出未標記中與起點最近距離之節點(節點 5)，其相鄰的點有節點 3(與起點距離 11)，因為 11 大於 9，所以不替換其與起點距離，其前節點也不更改。

判斷未標記中，節點 3 為與起點距離最短，將其標記。

節點	與起點距離	前節點
0(起點)	0	無
1✓	5	0
2	∞	尚無
3✓	9	1
4	17	1
5✓	8	0
6	∞	尚無
7	∞	尚無

第四步：

從上步驟所得出未標記中與起點最近距離之節點(節點 3)，其相鄰的點有節點 6(與起點距離 12)與節點 7(與起點距離 24)，因為 12 與 24 均小於 ∞ ，所以替換其與起點距離，並標記其前節點為 3。

判斷未標記中，節點 6 為與起點距離最短，將其標記。

節點	與起點距離	前節點
0(起點)	0	無
1✓	5	0
2	∞	尚無
3✓	9	1
4	17	1
5✓	8	0
6✓	12	3
7	24	3

第五步：

從上步驟所得出未標記中與起點最近距離之節點(節點 6)，其相鄰的點有節點 2(與起點距離 13)，因為距離 13 小於 ∞ ，所以替換其與起點距離，並標記其前節點為 6。

判斷未標記中，節點 2 為與起點距離最短，將其標記。

節點	與起點距離	前節點
0(起點)	0	無
1✓	5	0
2✓	13	6
3✓	9	1
4	17	1
5✓	8	0
6✓	12	3
7	24	3

第六步：

從上步驟所得出未標記中與起點最近距離之節點(節點 2)，其相鄰的點有節點 4(與起點距離 22)，因為距離 22 大於 17，所以不替換其與起點距離，其節點也不更改。

判斷未標記中，節點 4 為與起點距離最短，將其標記。

節點	與起點距離	前節點
0(起點)	0	無
1✓	5	0
2✓	13	6
3✓	9	1
4✓	17	1
5✓	8	0
6✓	12	3
7	24	3

第七步：

從上步驟所得出未標記中與起點最近距離之節點(節點 4)，其相鄰的點有節點 7(與起點距離 22)，因為距離 22 小於 24，所以替換其與起點距離，並更改其前節點為 4。

判斷未標記中，節點 7 為與起點距離最短，將其標記。

節點	與起點距離	前節點
0(起點)	0	無
1✓	5	0
2✓	14	6
3✓	9	1
4✓	17	1
5✓	8	0
6✓	12	3
7✓	22	4

3. 結果

所有節點均已被標記，得出下列結果

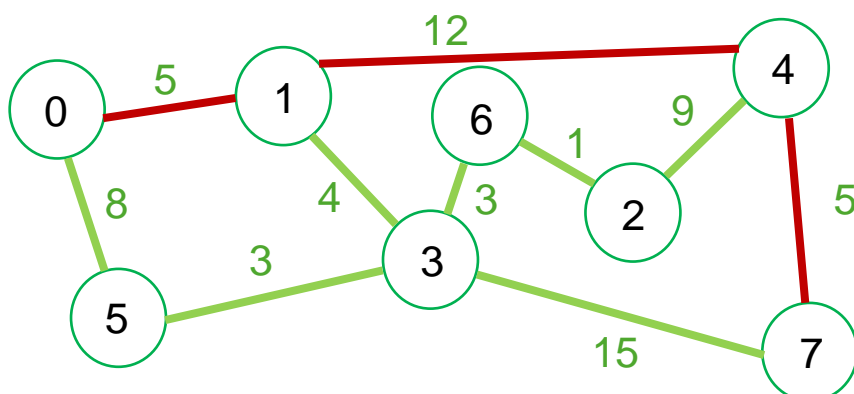
節點	與起點距離	前節點
0(起點)	0	無
1✓	5	0
2✓	14	6
3✓	9	1
4✓	17	1
5✓	8	0
6✓	12	3
7✓	22	4

至此所有節點與起點距離已判斷完畢，由前節點往回推，

節點 7(避難所) -> 節點 4 -> 節點 1 -> 節點 0(災害地)

得知從災害地到避難所最短距離為 22，路經由下圖所示(紅線)。

這條路徑可代表距離、時間或花費，將可提供救援隊作為判斷依據。



本學期的學習心得&課堂建議

本學期的學習心得

離散數學有別於一般數學(例如微積分、工程數學…等等)，對我來說，它不會有複雜的公式，但是推理能力必須要有，所以基本上不需要大量去記公式，學起來相比其他數學感覺更有意義了。因為離散數學不需大量公式，因此就算是其他領域的同學來學習也不會有太大的問題。

關於這學期的課程，一開始所介紹的邏輯、推論規則、命題等等，讓我覺得很抽象，沒有辦法讓我想像，我想是因為應用層面比較少或者我還沒接觸到，不過在學完整個課程之後，再藉由期中期末的報告，讓我可以了解各單元的應用層面。期中的部分針對課程前半部的單元，應用於實務專題上，期末報告針對後半部分的單元，由各同學自行尋找應用層面。期中報告讓我的專題變得更有興趣，並且可以用更簡單的方式向其他人說明，期末報告讓同學有更大的發展空間，而我也能將我喜歡的氣象套用上去，沒想到呈現效果還不錯(個人認為啦)，而且有些單元是完美運用到氣象學裡，例如演算法-數值天氣預報及圖-最短路徑算法，這時我終於了解離散數學的定位了，離散數學是一項工具，可以多樣化地應用於各個行業。

總結來說，離散數學讓我在繁重的課業下有一個喘息的地方，特別是這次的報告，讓我在課業中可以連結到我的興趣。謝謝老師這學期的指導，祝您事事順利，期待我們後會有期。

課堂建議

這堂課程的評分方式為期中期末報告，讓離散數學不會變成一個學科，而是真正能成為一個受用很久的工具，很喜歡老師這樣的安排。雖然每次幾乎都有作業，但都給同學足夠時間完成，如果稍微遲交，老師也很好說話。老師上課的方式也很有趣，會讓同學很認真地聽完，課堂的最後也都有習題可以練習，將此練習作為回家作業，現學現賣使我對課程內容更印象深刻。我也有注意到老師每天都會向同學們早安，讓我有一個很親切的感覺。

總之，我個人認為並沒有什麼需要改進的，老師很負責的完成課程的教授，期待未來還能上到老師的課程~