

## 資料結構-作業 4 報告

- 使用環境:

使用 Windows10 作業系統，並且使用 `code-blocks` 編譯器撰寫程式碼

- 遇到問題與困境，與解決辦法:

- ◆ **PROBLEM-1**

1. 遇到困境:整體而言還好，就是一開始要動態宣告陣列，然後把輸入的島嶼跟海洋矩陣讀取進去
2. 解決辦法:  
使用 `malloc` 語法，只是因為輸入的是二維，所以宣告動態陣列先用二為指標指向一維，再使用一維指標指到整數型態，用迴圈將他一一向記憶體要求空間存取，最後記得 `free`

- 程式內容:

主函式只是用來處理輸入跟輸出，重點在於宣告動態陣列，其他都是基本操作

`Numisland` 是用來遍歷所有矩陣元素，來看是否有島嶼，並且導入函式 `visit island`，為了判斷周遭(上下左右)是否是水域或是島嶼，來進而判定島嶼間相連，數目不需加一，反之若周圍是水域則是重新計算新的島嶼，最後回傳島嶼數

- 遇到問題與困境，與解決辦法:

## ◆ PROBLEM-2

1. 遇到困境:

當遇到無法形成完整連通圖的輸入值時候，需要判斷並輸出 **no connected**

2. 解決辦法:

原本就是判定用給定起點開始周圍的最小成本路徑，所以當遇到 0 成本(理論上不可能)，就代表有非連通情形，這時候我使用 **boolean** 值來看，如果遇到此情形就輸出 **no connected**

- 程式內容:

也是像第一題一樣動態宣告二維陣列，把輸入的 **adjacent matrix** 讀取最後要計算最短路徑總成本，根據起點開始後的節點，然後看周圍線路成本，判斷選擇的下一點，走過的存入父節點跟接下來存入 **key** 陣列，來方便我們觀察及選擇是否要加入此路線，並計算 **minimumcost**，還是換另一條路線，或者看到路徑 0 代表並沒有連通的節點，所以直接輸出 **no connected**

- 遇到問題與困境，與解決辦法:

### ◆ PROBLEM-3

- 1 遇到困境:非連通圖為特例需要討論

- 2 解決辦法:不同於 **prims algorithm**，由於選擇最小成本的邊開始建 **minimumcost**，所以可能要到最後才能知道是否有落單的邊存在，但是 **adjacent matrix** 還是要輸出，只是差別在於前面是否要印出 **no connected** 還是 **minimum cost**

- 程式內容:

建立資料結構 **graph** 還有頂點跟邊的結構，另外 **union,find,compare edges** 等函式分別解決建立最小生成樹的各步驟處理

把輸入的每一列第一位跟第二位分別表示兩頂點的連接，第三位表示相連邊的成本數，不論如何都要將 **adjacent matrix** 輸出，只是要判斷是否為連通圖，來決定是否要先輸出最小生成樹的 **minimum cost**，還是 **no connected**