

# 程式作業(五)

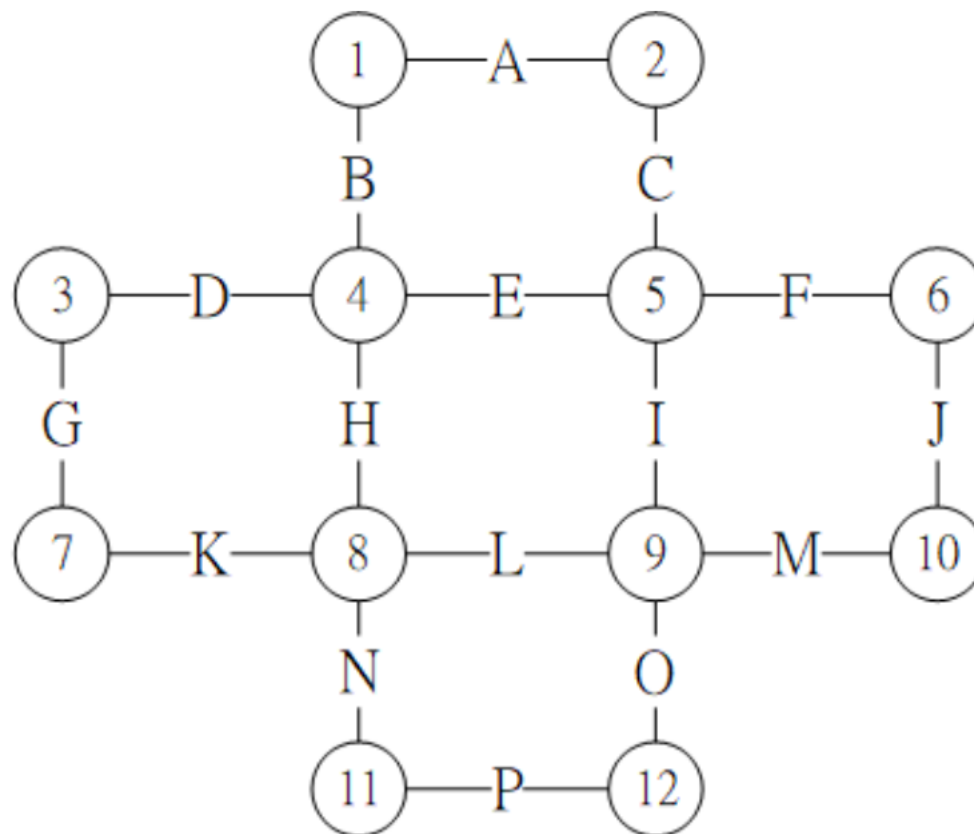
107520533

資工二A

張文耀

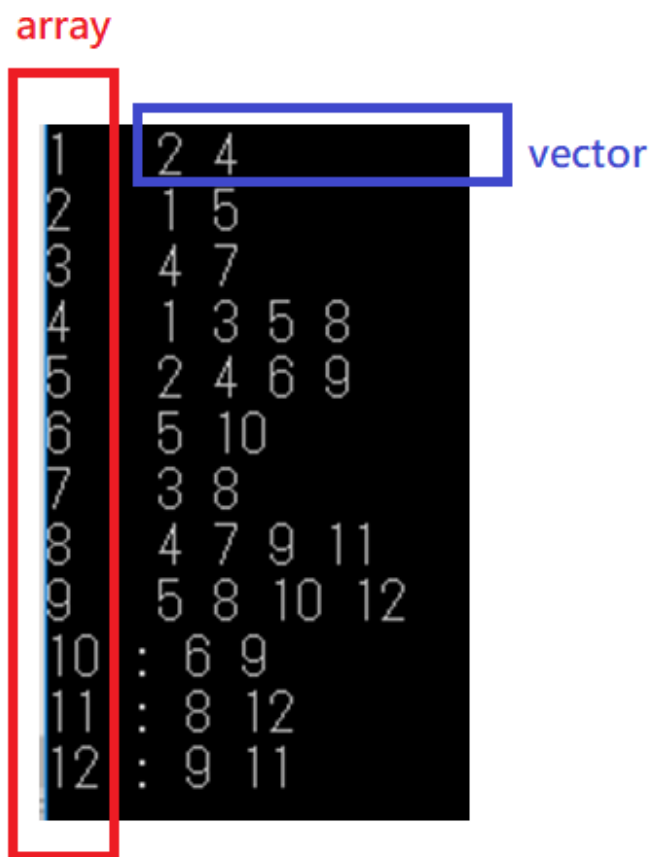
# 題目

給你一個圖，判斷圖是否可以一筆畫連起來，若可以，輸出圖。  
等等就用題目的圖當例子



# 建圖

我用adjacent-list，比較省時間，用一個vector的array去存圖，依照輸入的順序push進去，建出來像這樣



花費時間 $O(|V| + |E|)$

# 排序

因為要照字典序排列，我以我們要對list裡的所有鄰居由小到大排序，用quick sort排序。

花費時間 $O(|E| \log |E|)$

# 判斷

三種情況：

- 1.如果所有的vector的大小都是偶數，則從一號開始畫圖
- 2.如果其中兩個vector的大小為奇數，其他為偶數，則從編號較前面的奇數vector開始畫
- 3.如果超過兩個vector的大小為奇數，輸出”not exist”

花費時間 $O(|V| + |E|)$

# 畫圖

先建一個stack，依照順序走DFS，若走到死路，則把該點push進去stack，並把該邊封死。所有edge都走過一遍，每一遍再走各自的edge，所以

花費時間 $O(|E| * |E|)$

或是每次走過就直接把adjacent-list裡的edge刪掉，這樣子就會所有邊只跑一次，所以

花費時間 $O(|E|)$

*但是我太爛了，所以用第一個方法*

# 輸出

把stack裡的頂點輸出出來。

花費時間 $O(|E|)$

建圖 $O(|V| + |E|)$  排序 $O(|E| \log |E|)$  判斷 $O(|V| + |E|)$  畫圖 $O(|E| * |E|)$  輸出 $O(|E|)$

所以總體的時間複雜度為

$O(|V| + |E| * |E|)$

# Pseudo code

Determine(G) //判斷

for i = 1 to  $|V|$

if(sizeof(G.adj[i]) %2 ==1) //如果有奇數

counter++;

if(counter == 0 or counter == 2)

DFS(G, 1)

else

print("not exist")



# Pseudo code

Stack Circuit //畫過的圖

DFS(adj, v)

  for i=0 to adj[v].size()

    if(adj[v].at(i).visited == false)

      setTrue(v, adj[v].at(i).edge)

      DFS(adj, adj[v].at(i).edge)

  circuit.push(v)

# Pseudo code

```
Output(circuit)  //輸出  
  length = circuit.size()  
  for i=1 to length  
    print(circuit.top() + “ “)  
    circuit.pop()
```