

# My Notes

Important Concepts worth keeping

Today: / /

## 七橋問題 (seven bridge of Königsberg)

- $V(G)$ : vertex set 頂點
- $E(G)$ : edge set  $\rightarrow \sum \text{degree}(v_i) = |E(G)| \times 2$
- degree (Number of edges)
- 每個點可分為兩大類: <sup>odd</sup>奇數和<sup>even</sup>偶數
- 要符合沒有奇數 degree 或只有 2 個奇數 degree   
  $\rightarrow$  只可以這 2 種性質
- 如果只有 1 個奇數 degree? (3, 5, 7, ... 奇數都 No)   
  $\rightarrow$  因為一定是一進一出
- $\rightarrow$  可利用  $\sum \text{degree}(v_i) = |E(G)| \times 2$  證明   
  $\rightarrow$  degree 一定為偶數
- 0 個奇數 degree 走訪稱 Euler tour
- $\rightarrow$  開始和結束在同一個 vertex (點)
- 2 個點為奇數, 起點, 終點為奇數 (Euler walk)



## My Questions

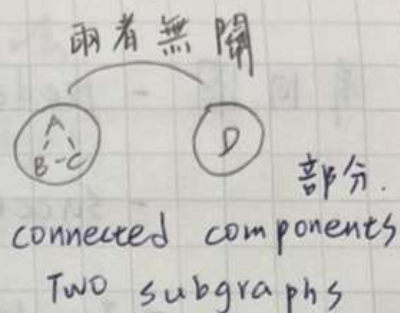
Problems & Difficulties needing exploration

## My Opinions

Thoughts, inspirations, and suggestions

### Basic Terminologies.

- connect graph. (無向圖)
  - 有 path (路徑在兩點之間)
  - 有可能 disconnected graph
- complete graph  $|E| = ?$  (無向圖)
  - 任兩人之間都有朋友關係
- strong connected graph (有向圖)
  - 任兩個點都有一條路徑存在
- weighted graph.
  - 權重, 把它放邊上比較好處理



密碼  
cipher key

如果你不知道你要去哪裡, 那麼現在你在哪裡一點都不重要。

《愛麗斯夢遊記》



- 有向圖 - predecessor 在前面的先行者
- successor 在後面的後繼者
  - In-degree 指進來的
  - out-degree 指出去的

## Adjacency Matrix 矩陣

- traverse 為  $O(V^2)$ , 點的個數平方成正比
- 沒有權重 0 或 1, 有權重記它的 weight (剩下記  $\infty$ )

## Adjacency List 串列

- traverse  $= O(V + E)$  點的個數 + 邊的總數

## Graph Traversals

- 圖中每個點都要走過
- 判斷是否為相連通
- 迴圈要解決, 利用 bool



## My Questions

Problems & Difficulties needing exploration

sequential representation 連續的表示法

- 利用 adjacency List (nodes + edges)

\*

P Q R S T W X Y Z

0 1 2 3 4 5 6 7 8

必須保留做區分

[0] [1] [2] [3] ... [8] [9] [10] [11] [12] [13] ...

10 12 13 14 20 20 2 5 6 6  
└─┘ └─┘ └─┘  
2 1 1

已∞)

$P \rightarrow R, P \rightarrow W \dots$  以此類推

→ 缺點：其中一個拿掉，就要全改

→ 最好不要做修改

→ undirected graph:  $|V| + 2|E| + 1$

## My Opinions

Thoughts, inspirations, and suggestions



# My Notes

Important Concepts worth keeping

Today: / /

DFS (Depth-First Search).

- 利用堆疊 (stack), last in, first out

BFS (Breadth-First Search)

- 利用 queue, first in, first out

- spanning tree 生成樹

→ 所有的點都涵蓋到, 邊的數量最少

△ 方法 ✓

- 利用迴圈寫好的 DFS (用 stack).



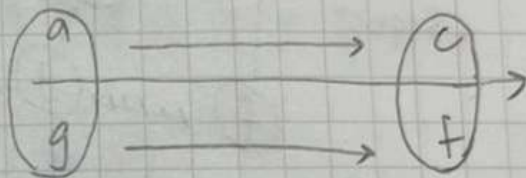
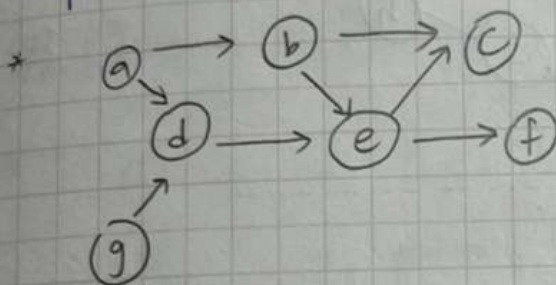
## My Questions

Problems & Difficulties needing exploration

### Topological Sort 拓撲排序

- 沒有權重，利用圖形結構
- 有向圖時使用，不可有 cycle
- Acyclic Digraph or Directed Acyclic Graph

簡稱 DAG



- 答案不是唯一，至少有一
- Activity-on-vertex (AOV) Network  
→ 以點為主

△ 方法 -

- ① 找 No successor 的點，後加入 beginning of a list
- ② 最後把和原本點相關的邊拿掉，repeat 3 步
- ③ 最終 graph 為空。

\* 不可以有 cycle

## My Opinions

Thoughts, inspirations, and suggestions

密碼  
cipher key

先從終點想回頭，再從起點開步走。



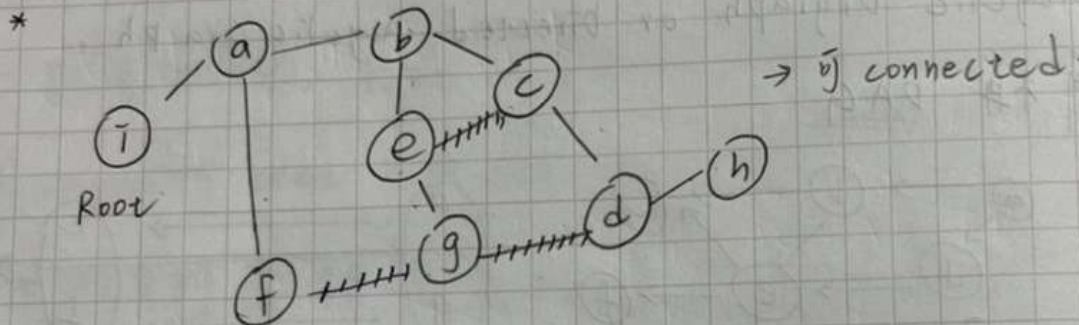
# My Notes

Important Concepts worth keeping

Today: / /

spanning Tree 生成樹

- 點和點都相連通，沒有方向性 (acyclic)  
(connect) (without cycle).



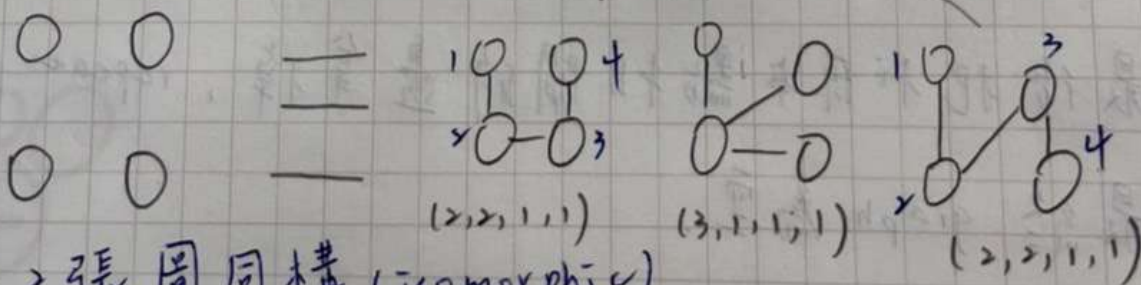
→ 移除直到沒有 cycle.

→ 假設有  $n$  個點，它一定有至少  $n-1$  個邊.

→ 只有多一個邊就會造成 cycle.

→ 少一個一定不是 connected.

◦ 有幾個不同 spanning Tree.



→ 多張圖同構 (isomorphic)

There is no disgrace in honest failure; there is disgrace in fearing to fail.

- Henry Ford



## My Questions

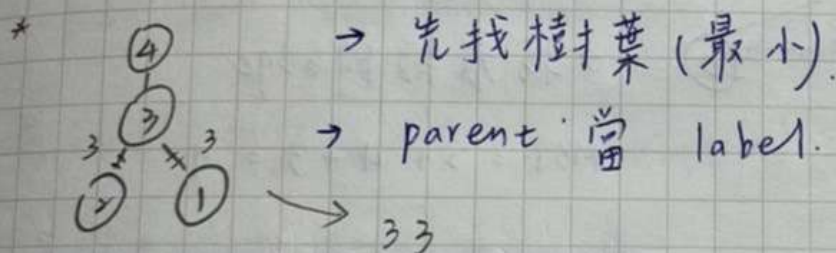
Problems & Difficulties needing exploration

有幾種可能? (公式  $n^{n-2}$ )

- 2 nodes  $\Rightarrow 2^{2-2} = 1$  00 -
- 3 nodes  $\Rightarrow 3^{3-2} = 3$  000 =
- 4 nodes  $\Rightarrow 4^{4-2} = 16$  0000 =

prufer sequence (普呂弗)

- 假設有  $n$  個點, 存下來的字串長度只有  $n-2$



- 字串是唯一的.

Minimum spanning tree  $\rightarrow$  利用區域

- 一張圖裡有多個生成樹, 會有一最小 (至少)

- 和 steiner tree - 樣.

-  $k$ -minimum spanning tree (指定幾個點).  
→ 整張圖  
數字

## My Opinions

Thoughts, inspirations, and suggestions

密碼  
cipher key



# My Notes

Important Concepts worth keeping

Today: / /

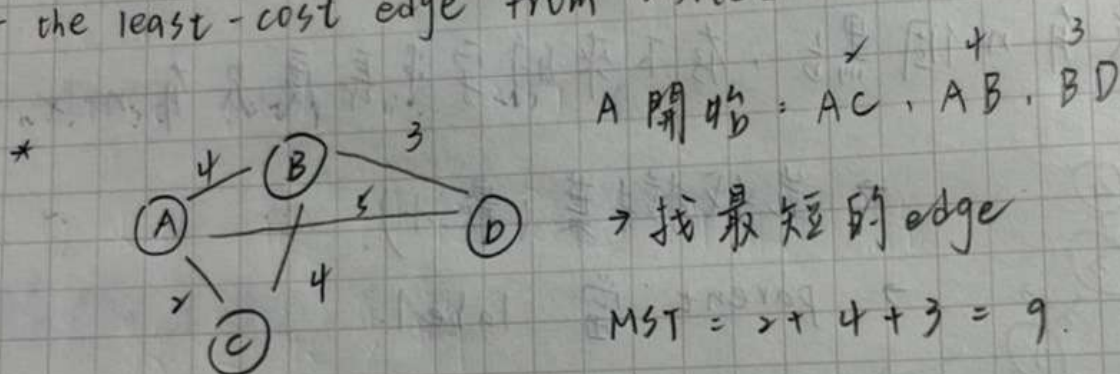
找 minimum spanning trees

△ 方法 - (Prim's Algorithm)

- 會找 cost 最少的邊，每個回合找一個，直到

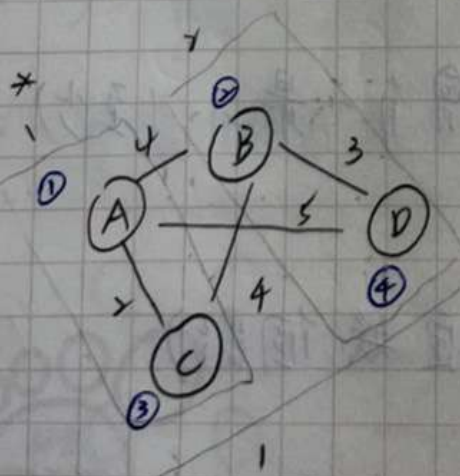
找到  $n-1$  個邊結束

- the least-cost edge from visited to unvisited



△ 方法 = (Kruskal's Algorithm)

- 每個回合合併兩棵樹，選最小 edge；直到  $n-1$



→ 有 4 棵樹

→ 先給 label，再檢查 label 是否一樣

第一回合 AC, BD, AB



## My Questions

Problems & Difficulties needing exploration

## My Opinions

Thoughts, inspirations, and suggestions

△ 方法三 (Sollin's Algorithm)

- 方法 = 的改進版
- 多一個 Edges, 存最短 edges.

## Shortest Paths

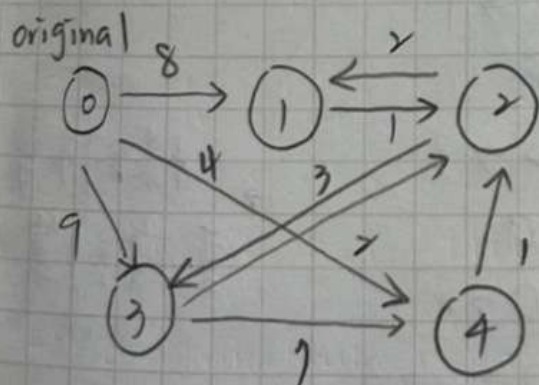
- 在頂點間, 最小的權重總和

→ between two vertices in a weighted graph is the path that has the smallest sum of its edge weights

△ 方法一 (Dijkstra's Algorithm)

- 可能有向圖可能無向圖

- 給一個起點<sup>original</sup>, 可找到所有最短路徑的點



為什麼選 4?

→ 找 minimum.

→ 找最少成本

→ 沒有轉機

step	v	vertexset	[0]	[1]	[2]	[3]	[4]
1	-	0	0	8	∞	9	4
2	4	0, 4	0	8	5	9	4
3	2	0, 4, 2	0	7	5	8	4
4	1	0, 4, 2, 1	0	7	5	8	4

密碼  
cipher key

少但是更好; 慢比快還要快。



# My Notes

Important Concepts worth keeping

Today: / /

## Dijkstra's algorithm

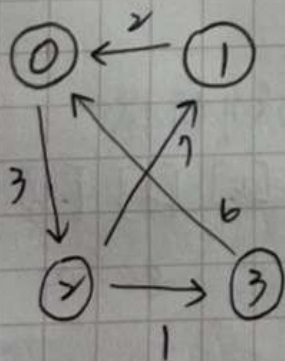
- 先把參數預設好
- 更新  $weight[u] = \min\{weight[u], weight[v] + edgeWeight[v,u]\}$
- 每個回合找最短:

## Shortest path vs. MST?

- 可以在一個圖裡涵蓋所有的點
- 邊數是最少的

## All-pairs shortest path

- 任何一個起點, 到任何一個點的最短路徑



0, 1	0 → 2 → 1
0, 2	0 → 2
0, 3	0 → 2 → 3
1, 0	1 → 0
1, 2	1 → 0 → 2
1, 3	1 → 0 → 2 → 3

→ 一樣的不要重複計算

All the roads worth walking are uphill.

- John Maxwell



## My Questions

Problems & Difficulties needing exploration

## My Opinions

Thoughts, inspirations, and suggestions

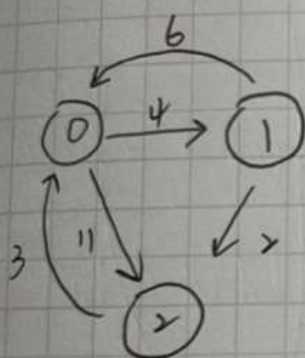
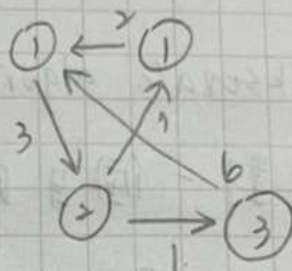
### Floyd's Algorithm (All pairs)

$D^0$ : all-pairs shortest paths 沒有轉折

$D^1$ : 在 0 轉折

$D^2$ : 在 0, 1 轉折

$D^3$ : 在 0, 1, 2 轉折



$D^1$	0	1	2
0	0	4	11
1	6	0	3
2	3	7	0

$D^0$	0	1	2
0	0	4	11
1	6	0	3
2	3	7	0

$D^2$	0	1	2
0	0	4	6
1	5	0	3
2	3	7	0

$D^1$	0	1	2
0	0	4	6
1	6	0	3
2	3	7	0

— 如果是無向圖，會是對稱的

密碼  
cipher key



## A\* algorithm

- Best-First search 路徑長越短越好
- 用和 Dijkstra's algorithm 一樣的資料結構  
→ 只針對一個目的地
- Greedy best-first search

$$f(v) = g(v) + h(v)$$

\* 猜測的

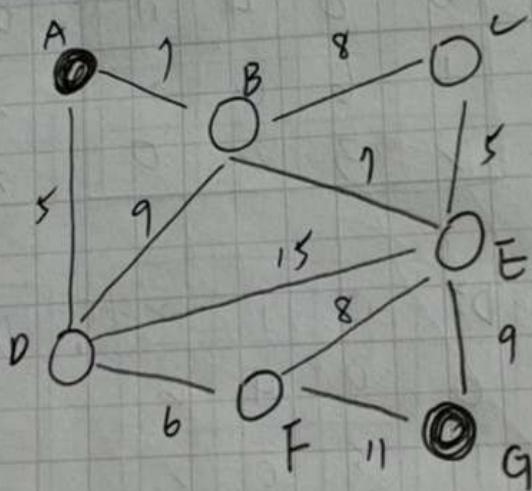
$$h(B) = 14$$

$$h(C) = 12$$

$$h(D) = 11$$

$$h(E) = 9$$

$$h(F) = 11$$



$$\Rightarrow f(B) = g(B) + h(B) = 1 + 14 = 15$$

$$f(D) = g(D) + h(D) = 5 + 11 = 16$$

$$f(C) = g(C) + h(C) = 1 + 8 + 12 = 21$$

$$f(E) = g(E) + h(E) = 1 + 1 + 9 = 11$$