演算法與資料結構（二）

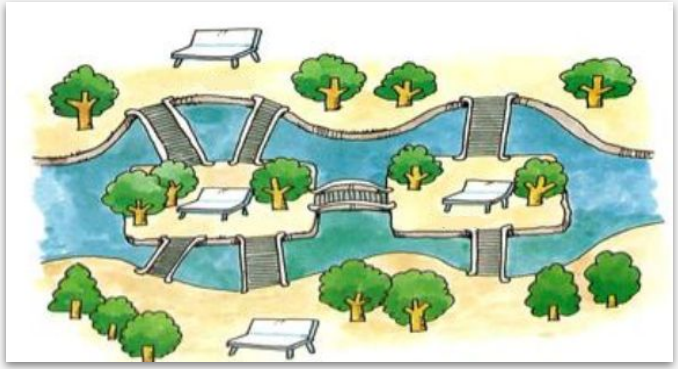
講者：大叫

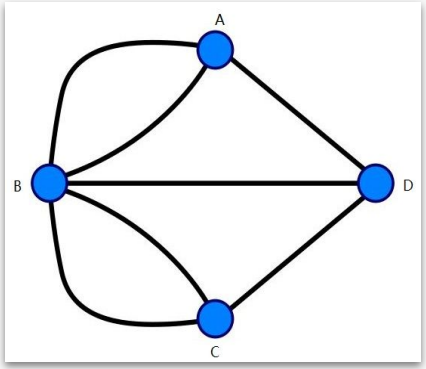
目錄

圖論

圖論

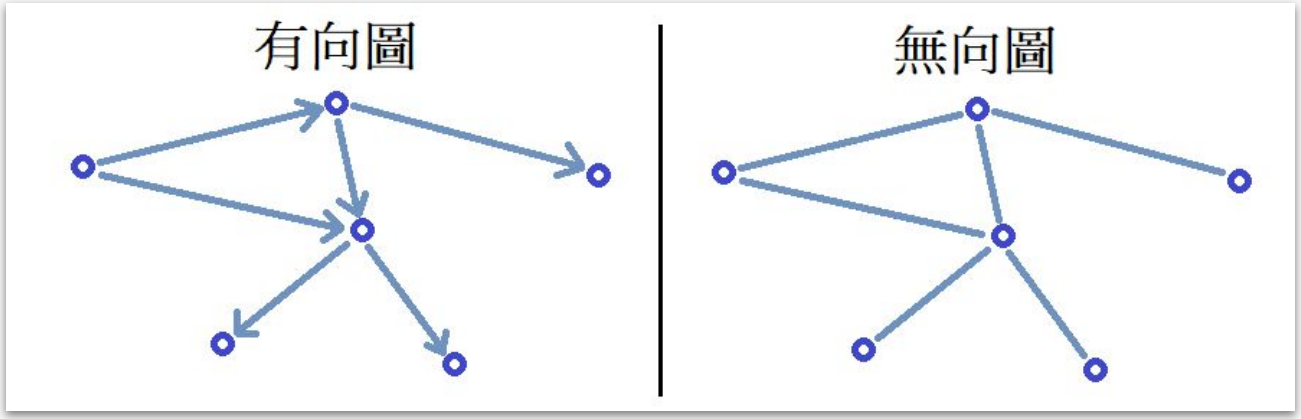
- 圖論是什麼？
- 主要研究圖（Graph）的性質、結構與應用的理論，圖由點（vertex）和邊（edge）組成，點描述事物，而邊是點之間的某種關係
- 可以把一些問題轉變為圖，再用圖論方法（如某種演算法）解決
- 經典例子：七橋問題





圖論

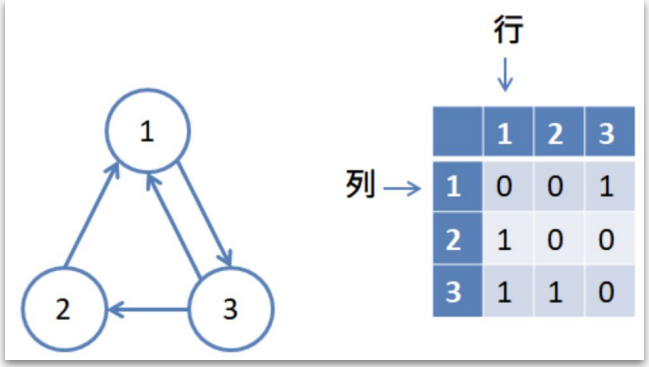
- 有向/無向圖：差別在邊有沒有方向性，有向的邊只能單向通過。
- 生活中有什麼類似的例子？



- C++ 中通常如何儲存邊的資訊？

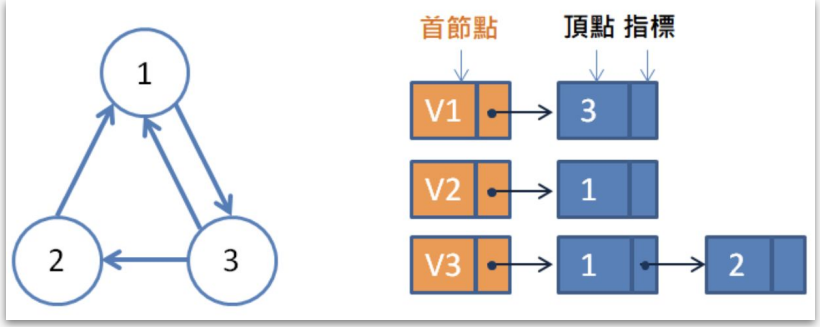
相鄰矩陣 (Adjacency Matrix)

```
int edge[vertex_num][vertex_num];

void add_edge(int u, int v) {
    edge[u][v] = 1;
}
```



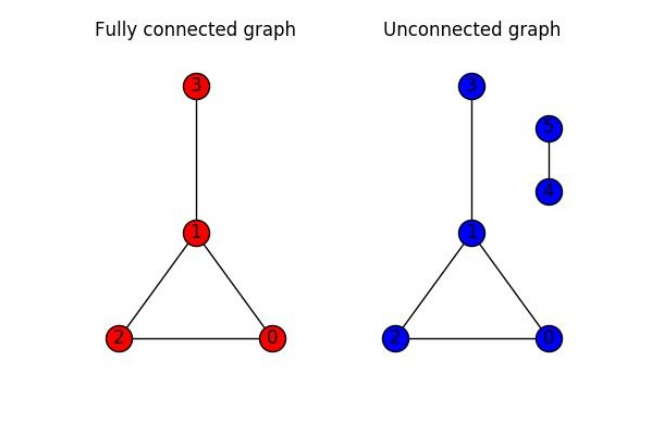
相鄰串列 (Adjacency List)

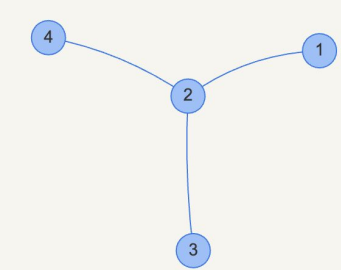
```
vector<int> edge[vertex_num];

void add_edge(int u, int v) {
    edge[u].push_back(v);
}
```

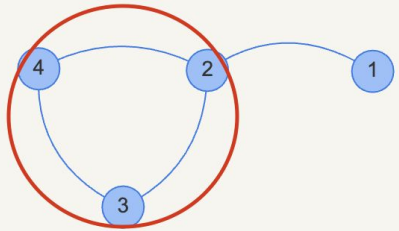


圖論

- 連通：若兩點連通，代表兩點間可以透過至少一條邊來抵達對方
- 連通圖：整張圖的點都相互連通的圖
- 環：一張圖有環，代表有一個走法，可以在經過一些不重複的邊後，走回起點

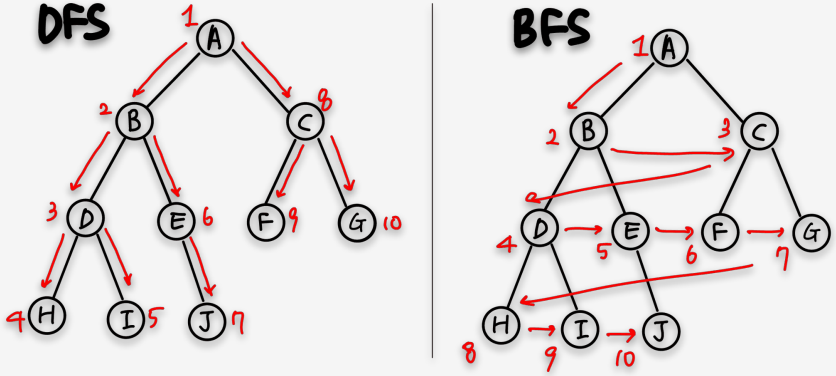






圖論

- 圖的走訪：
- DFS (Depth-First Search,DFS) 與 BFS(Breadth-First Search)
- DFS 就像在走迷宮一樣，選擇一條路一直走，直到走到死路，就往回走到還不是死路的地方
- BFS 就像在倒水，從起點開始慢慢的向外擴散
- 太抽象了吧==



圖論

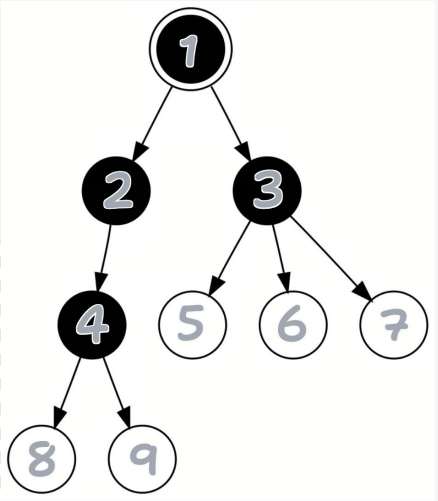
- 例題：CSES Counting Rooms
- 給一二維地圖，# 代表牆壁，. 代表地板，問該地圖中有幾個相互連通的區塊
- $1 \leq n, m \leq 1000$

- 作法

樹

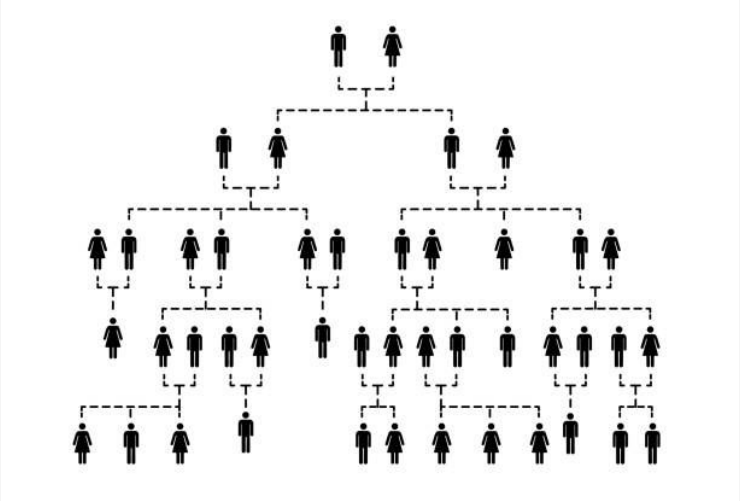
樹

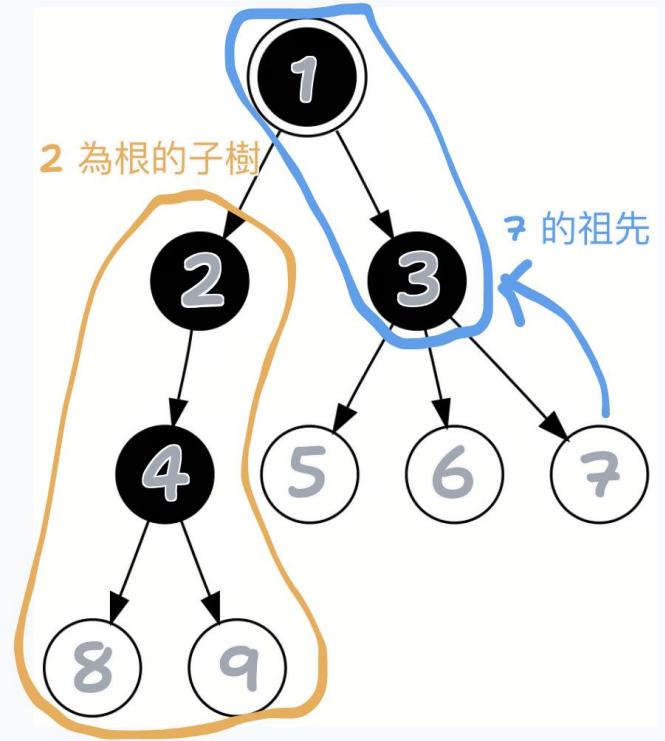
- 樹是什麼？
- 沒有環的連通圖！會發現邊的數量一定是點的數量-1
- root：一棵樹的根，唯一一個沒有父節點的節點（位於最上方
- x 的父節點：x 的上層節點
- x 的子節點：x 的下層節點（們
- 葉節點：沒有子節點的節點（位於最下方
- 資工系的樹，根在上葉在下！！！



樹

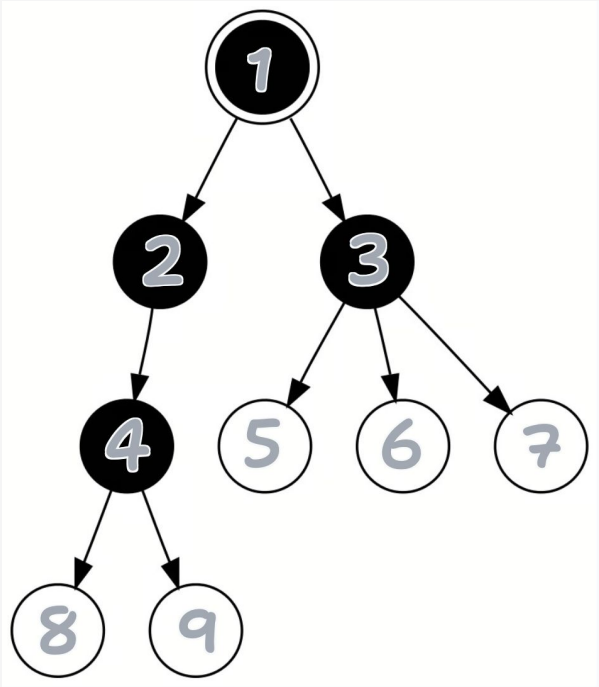
- x 的祖先：x 到根節點的路徑上的節點
- x 的後代：所有以 x 為祖先的節點
- 以 x 為根的子樹：x 與他的所有後代





樹

- 回答 slido 上的問題

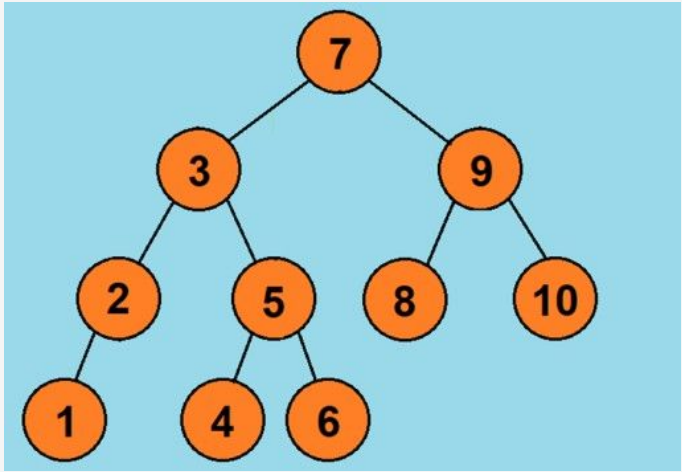


問題連結

二元樹與遍歷

二元樹與遍歷

- 二元樹是什麼？
- 最經典的一種樹，非常多資料結構都是以二元樹為基底
- 每個節點的子節點一定只能是左或右子節點，所以至多會有兩個子節點（也可以都沒有或只有其中一個）



二元樹與遍歷

- 遍歷在做什麼？
- 透過指定的規則，來依序走訪節點，不同的方式有不同的使用場景
- 中序遍歷 (Inorder Traversal) - LVR
- 前序遍歷 (Preorder Traversal) - VLR
- 後序遍歷 (Postorder Traversal) - LRV

- L、R、V 分別代表遍歷左子樹、遍歷右子樹和得到當前節點的資訊，只要記住每種遍歷的順序，就可以輕易的模擬遍歷的過程與結果。
- 模擬看看

二元樹與遍歷

- 有沒有辦法只用某兩種遍歷方式的結果，來得到整棵樹的結構？
- 其實可以！！！ 練習題

內容回顧

- 圖論
- 樹
- 二元樹與遍歷

QA

- 例如：
- 對前面的內容有哪裡想問？
- 其他

謝謝各位

講者：高睿