

Important Words

vertex --edge--> vertex : directed

vertex <--edge--> vertex : undirected

Adjacency Matrix

A --edge--> B : $[A][B] = 1, [B][A] = 0$

A <--edge--> B : $[A][B] = 1, [B][A] = 1$

A <--edge-- B : $[A][B] = 0, [B][A] = 1$

A B : $[A][B] = 0, [B][A] = 0$

Adjacency List : Graph define Set, so the order of vertex is not important.

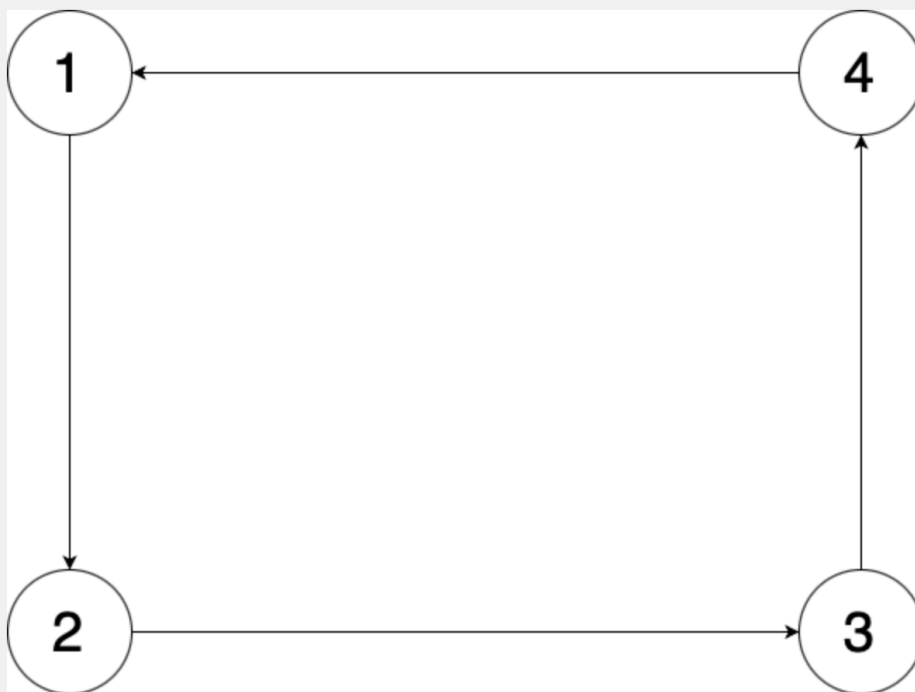
undirected : List out all elements, and linked all the vertexes that linked to them.

directed : Use the pointers to define which to linked to the elements.

•

- Cycle

- 不會重覆經過同一個點和同一條邊的路線，頭尾循環。

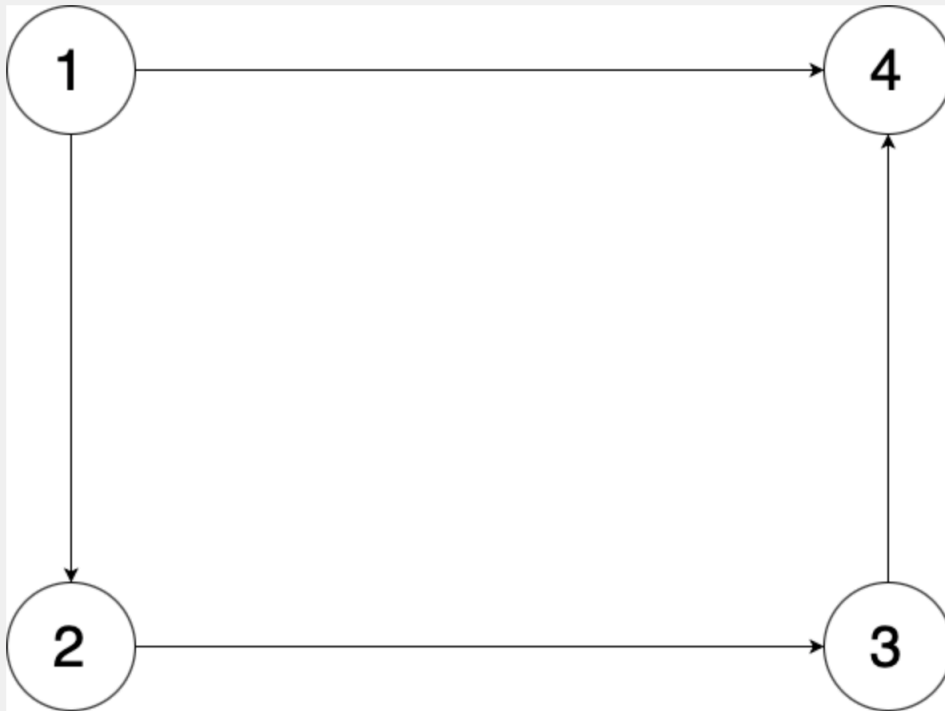


從上可見：

1. 由 4 走到可以走到 1，但 1 不能走向 4
2. 由 1 走到可以走到 2，但 2 不能走向 1
3. 由 2 走到可以走到 3，但 3 不能走向 2
4. 由 3 走到可以走到 4，但 4 不能走向 3

由此可見，此為一個「有向」且「有環」的圖

在此舉一反例：「有向無環圖」



1. 由 1 走到可以走到 4，但 4 不能走向 1
2. 由 1 走到可以走到 2，但 2 不能走向 1
3. 由 2 走到可以走到 3，但 3 不能走向 2
4. 由 3 走到可以走到 4，但 4 不能走向 3

* 不能從 3 到 4 再到 1 的循環 *

補充名詞定義：

simple trail: 邊不可重複(頂點可能重複)，例如：公車可以走不同路線到同一站牌。

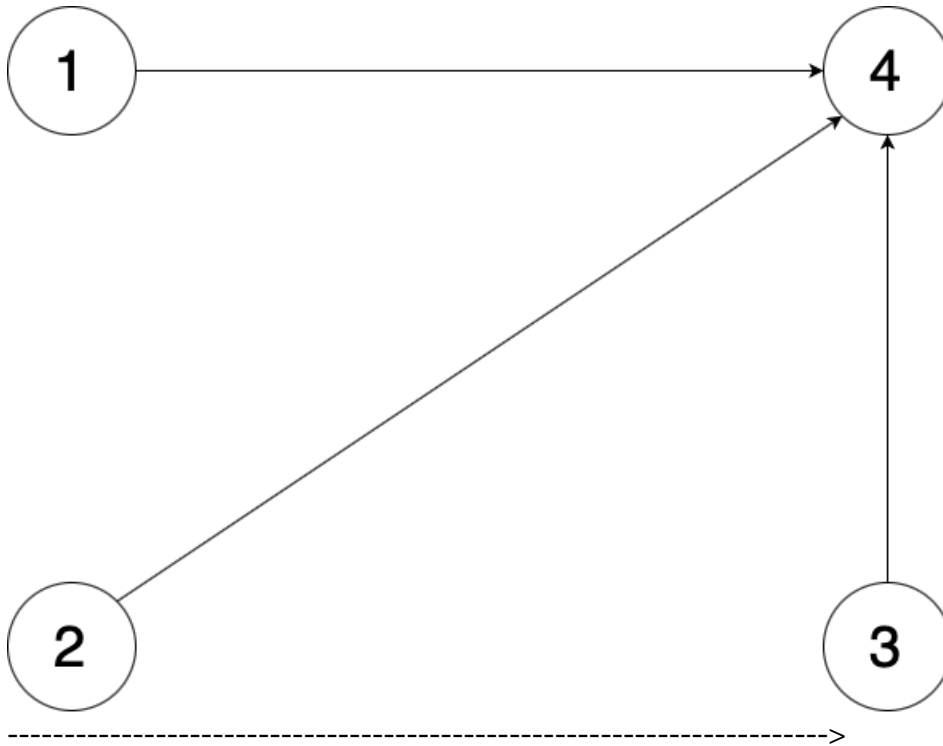
simple path: 頂點不可重複(邊也不可能重複)，例如：旅遊路線不適合舊地重遊吧！

circuit: 首尾同一頂點的simple trail(邊不可重複，但頂點可能重複)，例如：路跑可在路線重疊點放置補充站。

cycle: 首尾同一頂點的simple path(邊和頂點都不可重複)，這就是大家熟知定義的循環。

- Directed Acyclic Graph (DAG)
 - 有向無環圖
- Topological Sort
 - 拓撲排序
 - 只會由 a 點指向 b 點，b 點不會指向 a 點 // 例如：「有向」且「無環」圖。
 - 一張圖可以有很多種拓撲排序，只要不違背圖上每一條邊的先後順序，怎麼連都可以。

- 圖上 不可以 有環（ cycle ）！
- 圖解：



此圖為有向無環圖

從上可見：

由 1 走到可以走到 4，但 4 不能走向 1

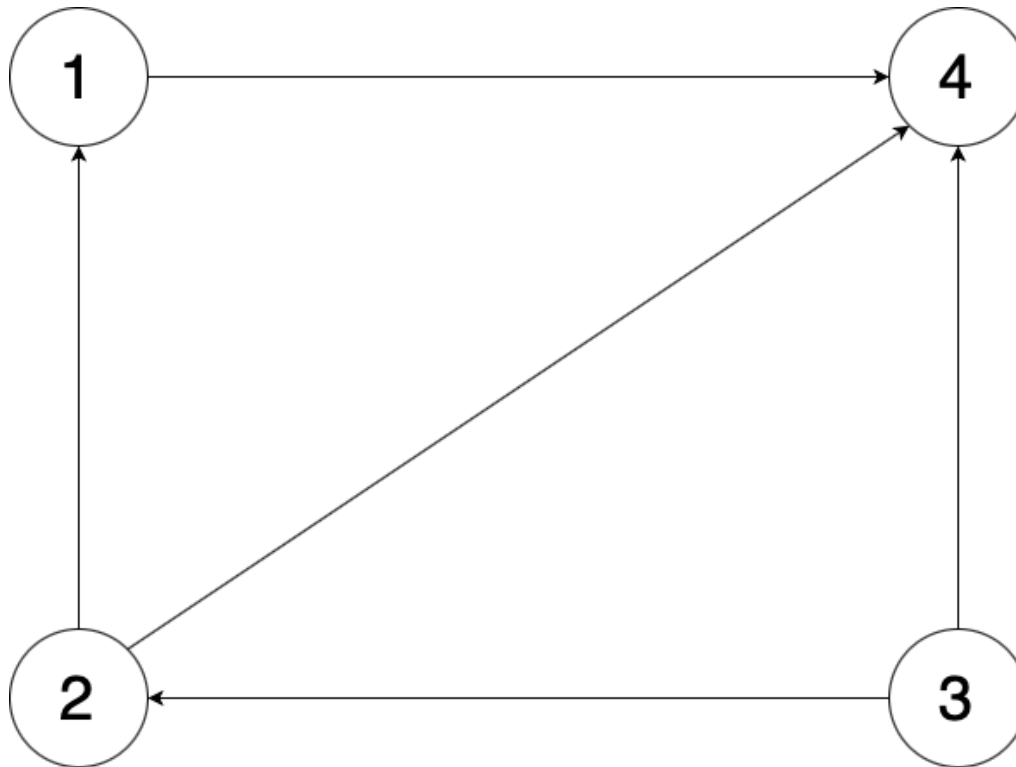
由 2 走到可以走到 4，但 4 不能走向 2

由 3 走到可以走到 4，但 4 不能走向 3

* 3 / 4 個 vertex，都同一指向 vertex（ vertex4 ） ，並且無法從 vertex4 往回到 vertex1, 2, 3 *

從上可見一重要重點：當有向圖出現循環(cycle)，就無法獲得拓撲排序

排序方法說明：



用陣列來表示，每個 vertex 儲存的 edge 數量：

step 1：初始化歸零

vertex 1：0

vertex 2：0

vertex 3：0

vertex 4：0

step 2：數數量

vertex 1：1

vertex 2：1

vertex 3：0

vertex 4：3

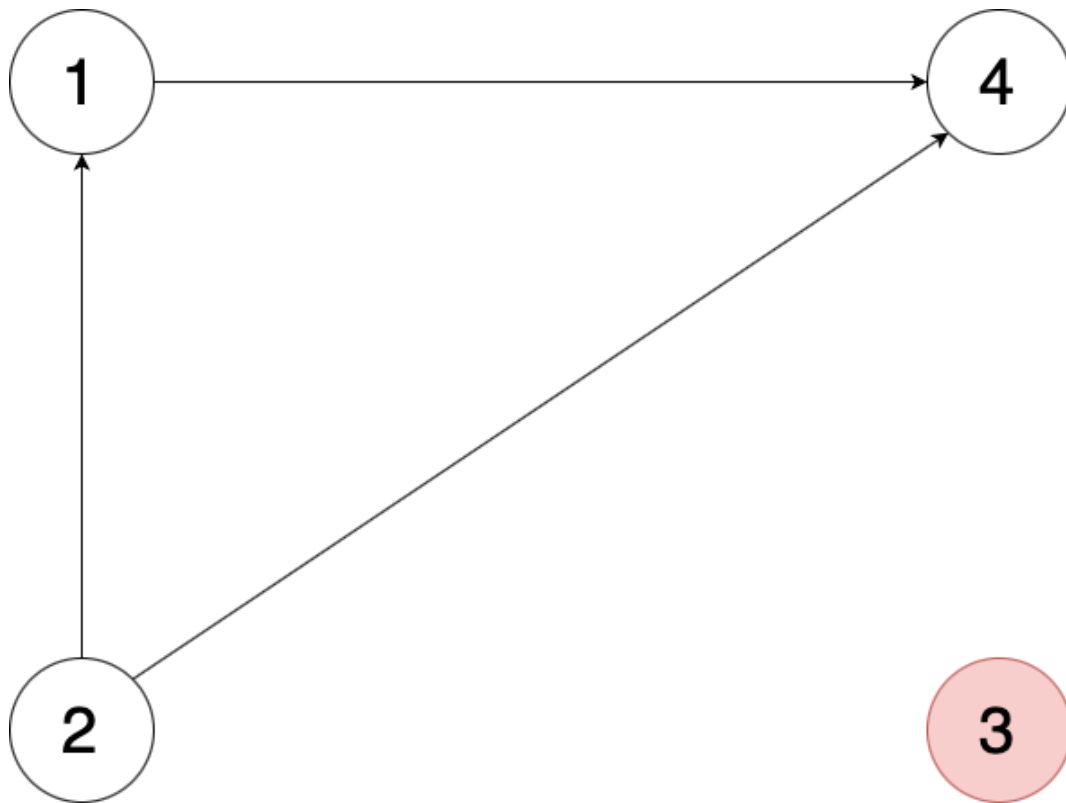
step 3 : 刪除 edge 數為 0 的 vertex

vertex 1 : 1

vertex 2 : 2

vertex x : x

vertex 4 : 3



step 4 : 重新計算當前 vertex 的 edge

vertex 1 : 1

vertex 2 : 0

vertex x : x

vertex 4 : 2

step 5 : 刪除 edge 數為 0 的 vertex

vertex 1 : 1

vertex x : x

vertex x : x

vertex 4 : 2

變成下圖



step 6 : 重新計算當前 vertex 的 edge

vertex 1 : 0

vertex x : x

vertex x : x

vertex 4 : 1

step 7 : 刪除 edge 數為 0 的 vertex

vertex x : x

vertex x : x

vertex x : x

vertex 4 : 1

變成下圖



step 8 : 重新計算當前 vertex 的 edge

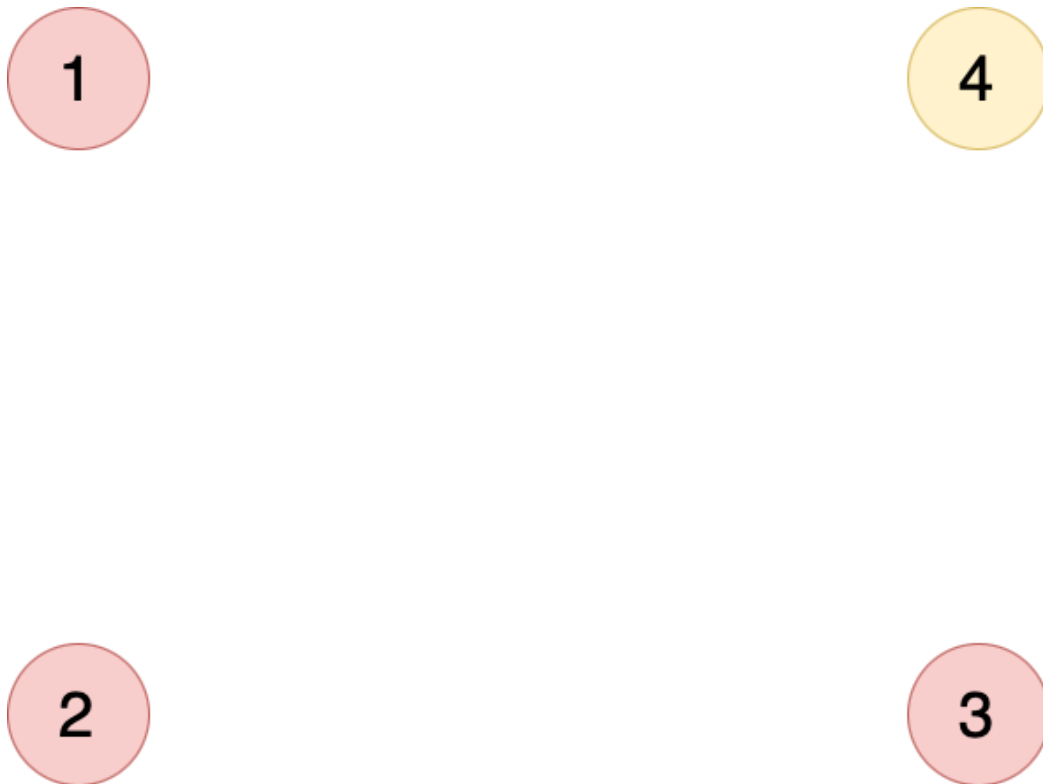
vertex x : x

vertex x : x

vertex x : x

vertex 4 : 0

最後變成下圖



此次範例的拓撲排序為：vertex 3 --> vertex 2 --> vertex 1 --> vertex 4

因為拓撲排序的排序方式為「把 edge 為 0 的 vertex 先刪除」，所以不能是循環cycle，否則將會無法刪除 vertex

- Spanning Tree
- 「生成樹」
- 從一張圖取出一棵樹，包含圖上所有點。
- Minimum Spanning Tree
 - 權重最小的生成樹。
- Shortest Paths
 - 「最短路徑」是由起點到終點、權重最小的路徑。
 - 可能有很多條。
 - 起點到終點不通、不存在路徑的時候，就沒有最短路徑。
- Critical Path Analysis (CPM)
 - 關鍵路徑法

- 為了表示 performance 而存在的。
- Maximum Flow Problem
 - 最大流
 - 在一張圖中，以及給定一個源點與一個匯點，所有可能的 Flow 當中，流量最大者就是 Maximum Flow。
 - 可能會有很多個。
- B-Tree
 - Balance Tree
 - 平衡的多路搜尋樹
 - 根節點要就是葉子, 不然至少有兩個子節點。
 - 每個節點最多含有m個節點($m \geq 2$)。
 - 所有的葉節點都出現在同一層上。
- External Sort
 - 外部排序
 - 分次載入部份的資料到記憶體，用內部排序演算法排序後再回存或合併結果。