Graph Algorithms
Topological Sort

for Sprout 2014 by Chin Huang Lin

Sprout

回憶一下……

- 圖中的邊可以對應到兩個元素之間的「關係」
- 樹中的邊可以對應到兩個元素之間的「輩分」
- 邊有分有向與無向，分別代表單向與雙向的「關係」
- 無向樹的輩分會存在，來自於根據樹根做的定向
- 那麼類似地，在有向圖上能不能夠也有「輩分」之分呢……？

問題來也

- 有向圖上有 n 個點，m 條邊
- 假如把點依序編號為 $1\sim n$，存不存在一種排列 $p_1, p_2, \ldots, p_n$，滿足對於任意一條邊 $x \to y\,(x = p_i, y = p_j)$，都有 $i < j$ 呢？
- 我們稱滿足這樣條件的序列為一個 *Topological Order*

觀察一下

- 對於當前的圖，假如
  - 一個點還有任何的入度，那麼它一定不是序列的第一個元素
  - 一個點沒有任何的入度，那麼選它當第一個元素肯定不會出事！
- 決定好第一個元素後，與第一個元素相關的限制就都不要緊了
  - 所以可以把該元素的所有出度都拿掉
- 剩下的部份不管順序如何都與第一個元素再也無關
  - 是個完全獨立的問題
- 可以遞迴處理！

算法概念

對於一張圖 $G = \{V, E\}$

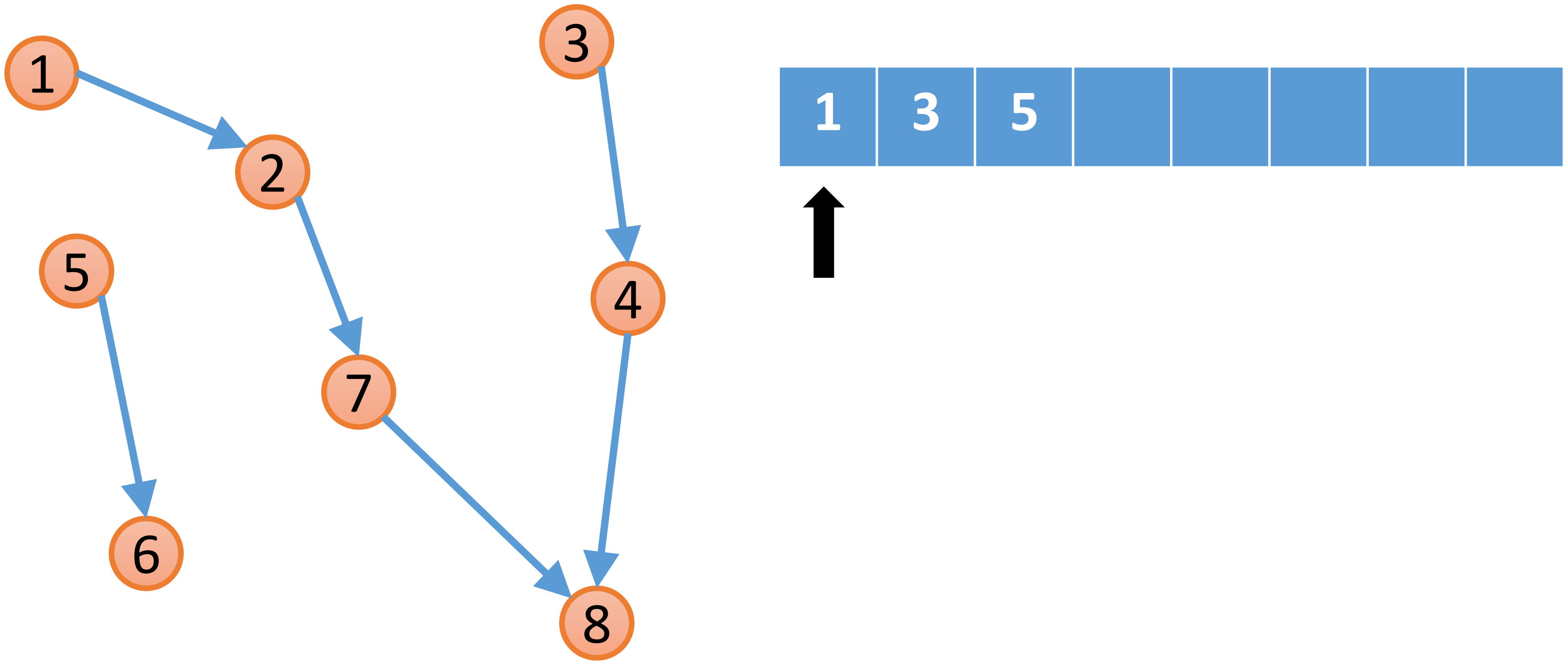
1. 找到一個入度為 0 的點 p
2. 把 p 和它的出度都拔掉，形成 G'
3. 遞迴求解 G' 的 topological order，然後接在 p 後面形成 G 的 topological order

- 實作上我們當然不需要真的遞迴下去
- 用一個 queue 加速算法的第一個步驟

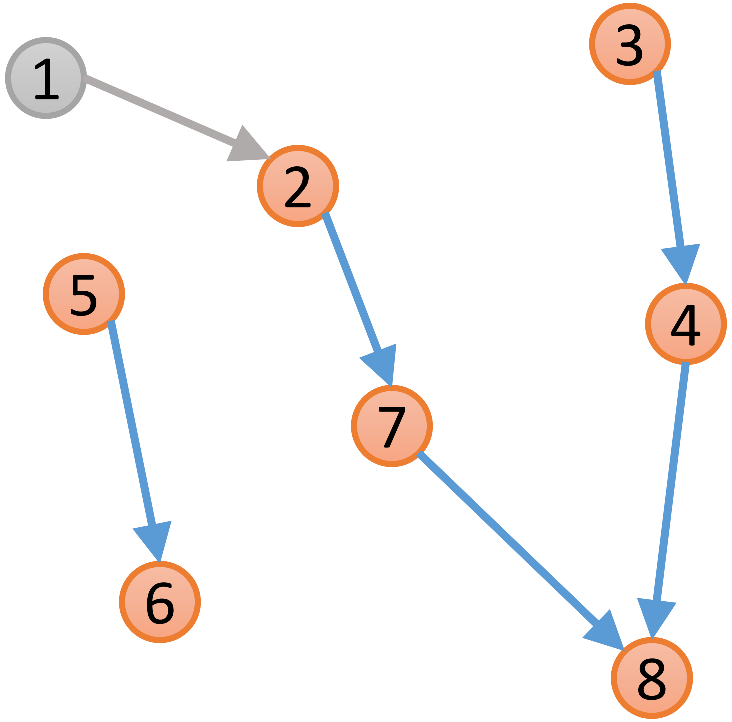
方法一：BFS 變形版本

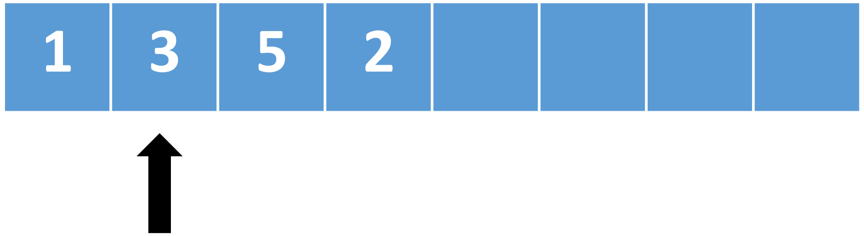
1. 初始將所有入度為 0 的點都推入 queue
2. 從 queue 中取出元素 p
3. 將 p 的出度都移除掉，並維護各個點的入度值
4. 如果某點在上步驟執行後入度變為 0，則將該點推入 queue
5. 若 queue 不為空，回到步驟 2
6. 算法結束時，取出的順序即為一組解

方法一：BFS 變形版本

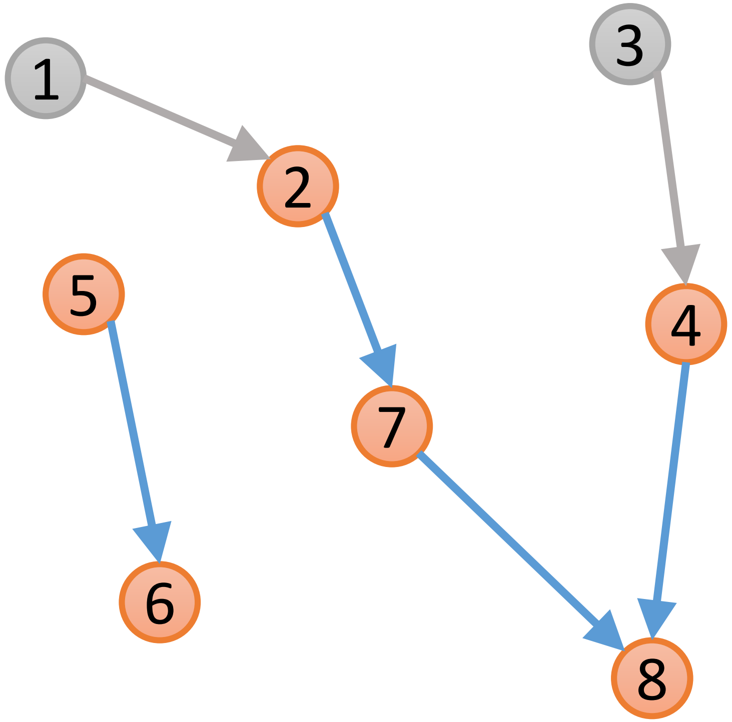


方法一：BFS 變形版本





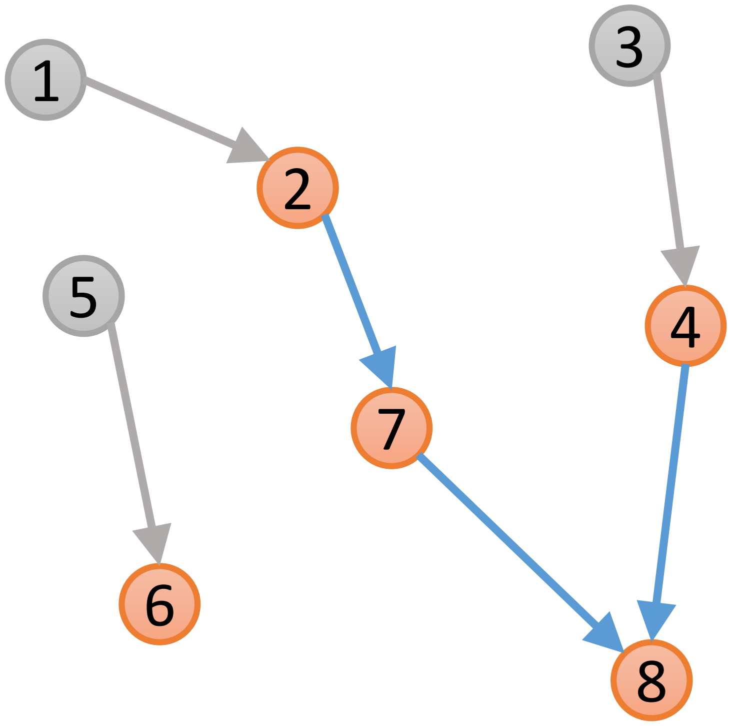
方法一：BFS 變形版本



1	3	5	2	4			

Sprout

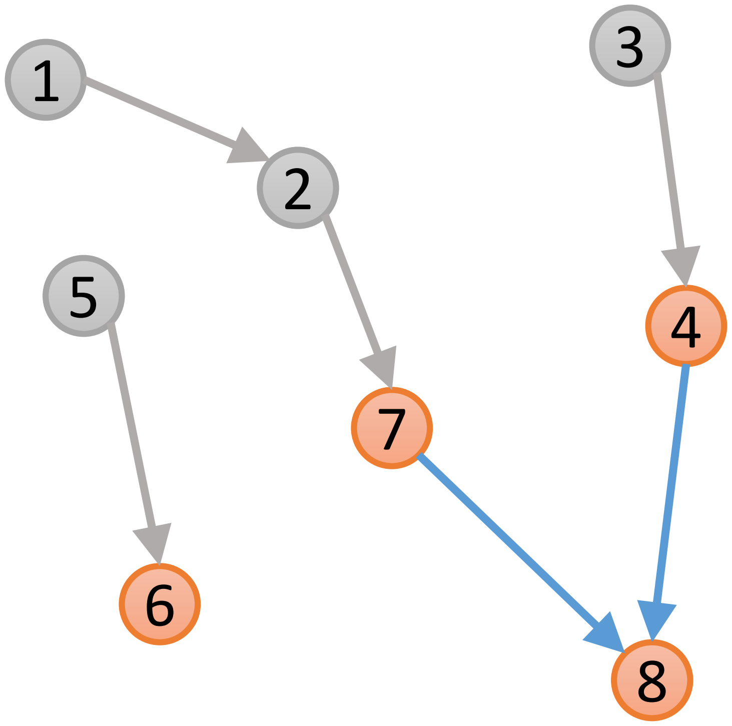
方法一：BFS 變形版本



1	3	5	2	4	6		

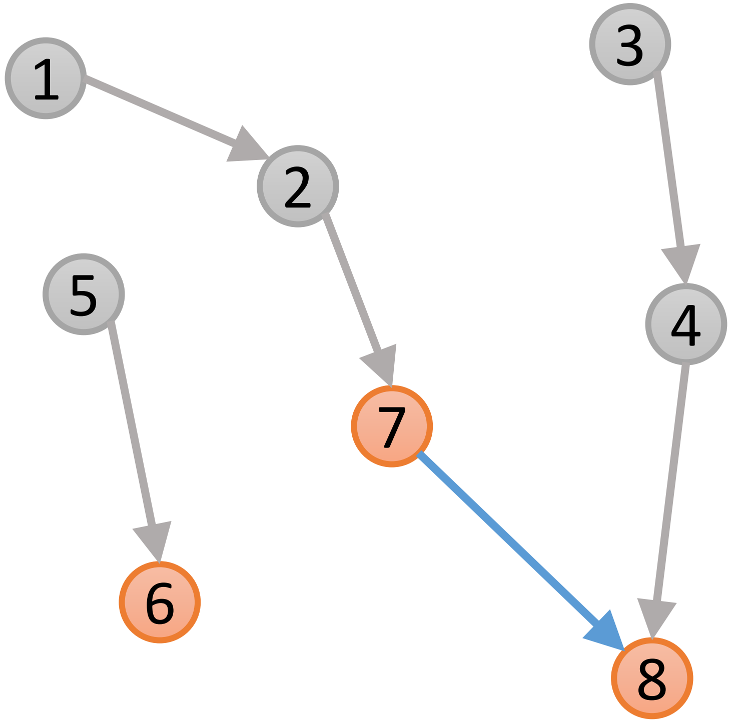
↑

方法一：BFS 變形版本



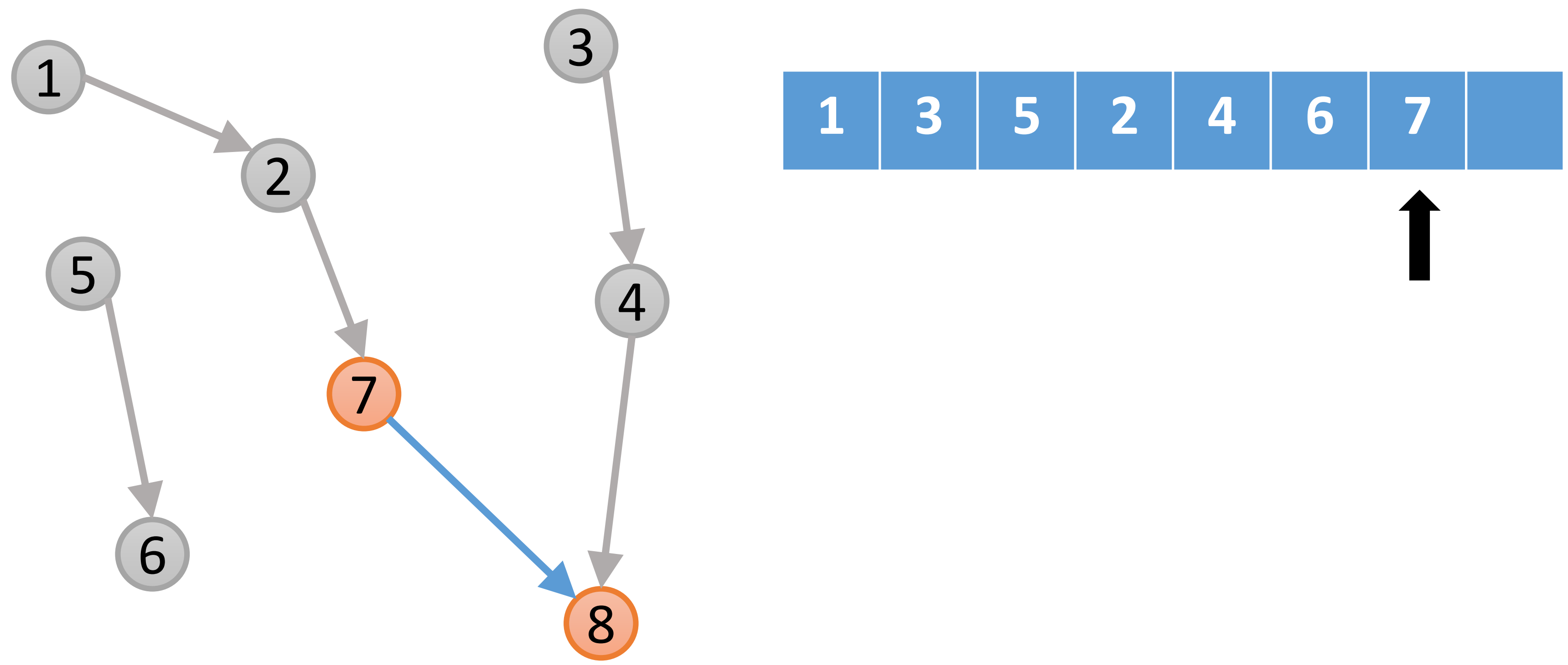
1	3	5	2	4	6	7	

方法一：BFS 變形版本

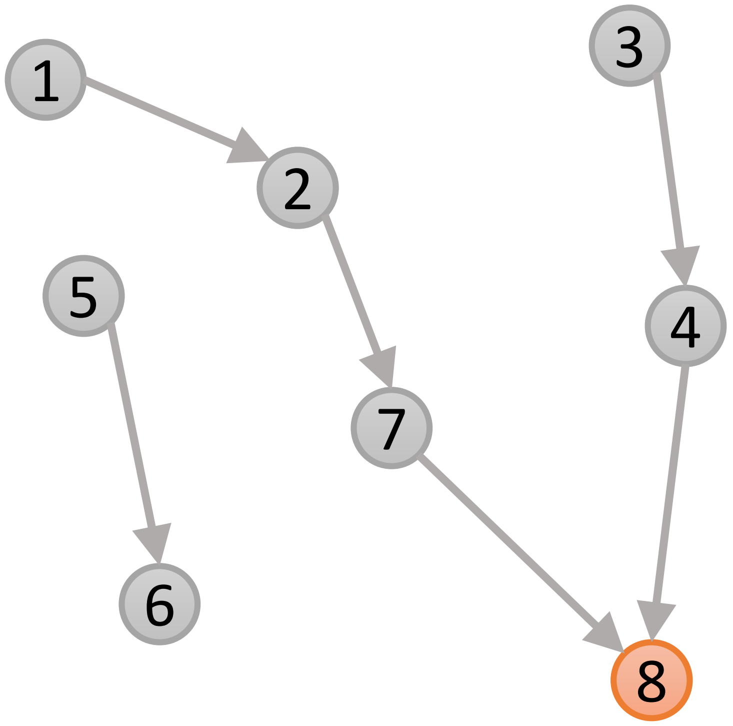


1	3	5	2	4	6	7	

方法一：BFS 變形版本

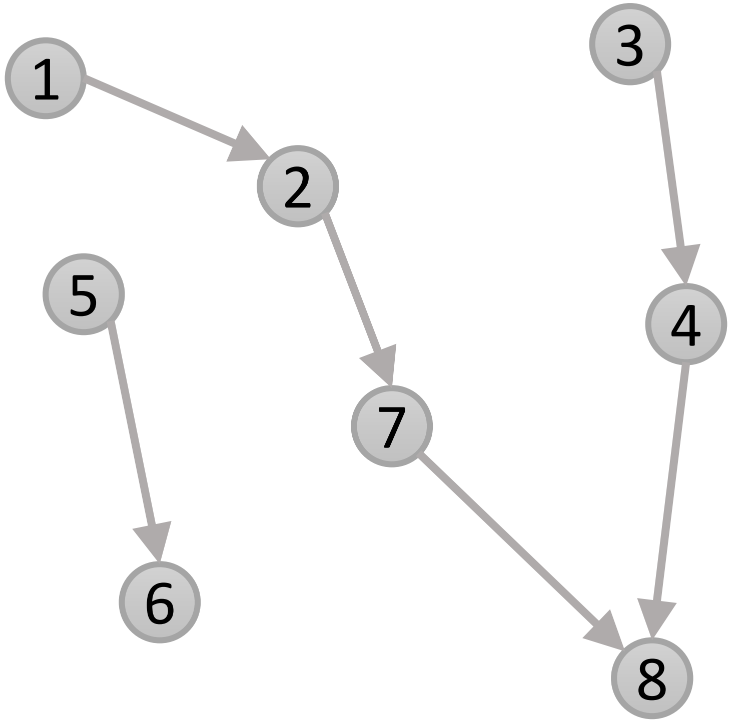


方法一：BFS 變形版本



1	3	5	2	4	6	7	8

方法一：BFS 變形版本



1	3	5	2	4	6	7	8

會不會有特例呢……？

- 如果算法給出一組解，這組解一定合法
- 如果算法給出的不是一組解，只有一種可能
  - 步驟 5 中，queue 已經空了，但是還有點還沒有被拜訪過！
- 此時代表圖中剩餘的點必定都至少有一個入度
  - 也就是說，圖上還剩下至少 n 條邊

→ 圖上必定存在環 (cycle) (想一想，為什麼？)

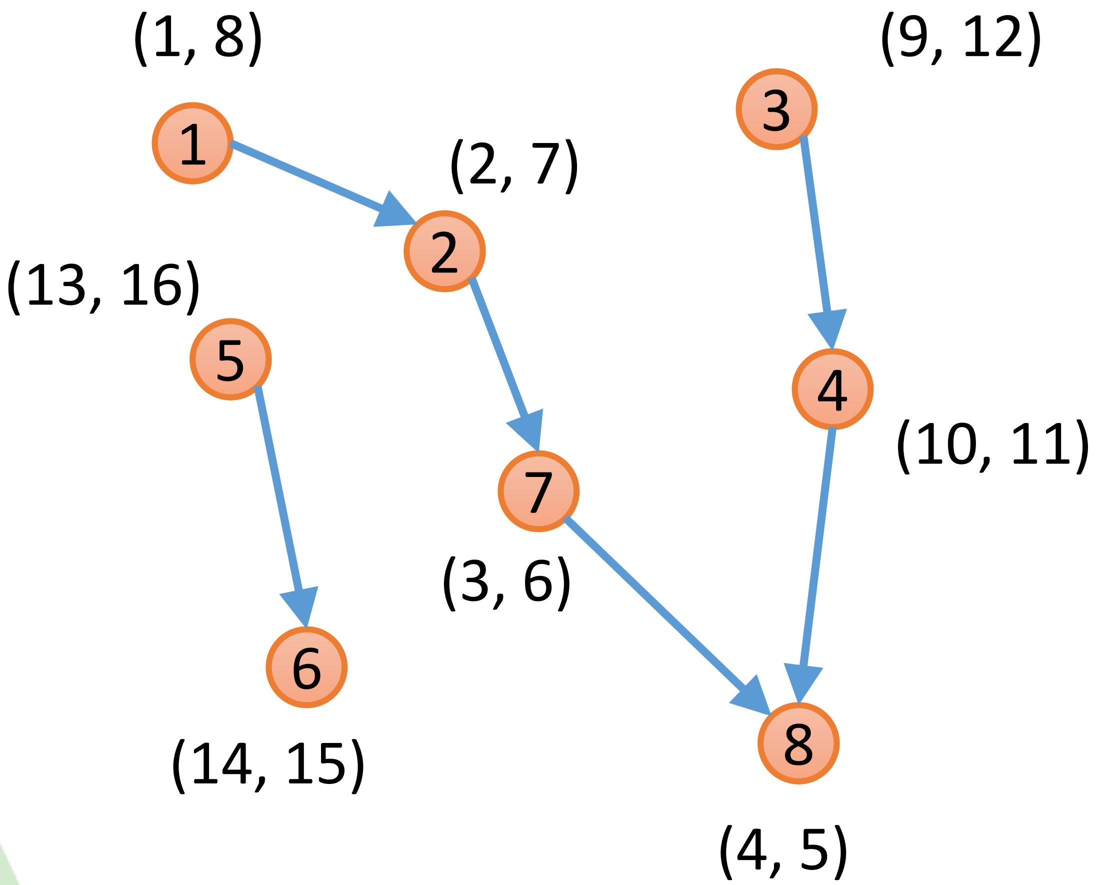
官官相護何時了

- 在一個有環的圖上，不可能存在 topological order
  - 否則至少順序最前面的元素會不合法
- 一張圖如果有向而且沒有環，我們就稱之為有向無環圖 (DAG, directed acyclic graph)
- Topological Sort 其實順便完成了 DAG 判定~

另一種想法

- 第二種想法的關鍵概念是時間戳記 (time stamp)
- 在 DFS 的過程中，我們會
  - 進入一個點
  - 在該點停留一段時間 (這段期間會拜訪所有該點可及的子孫們)
  - 離開該點
- 假如我們在進入一個點和離開一個點時分別留下戳記的話……

時間戳記

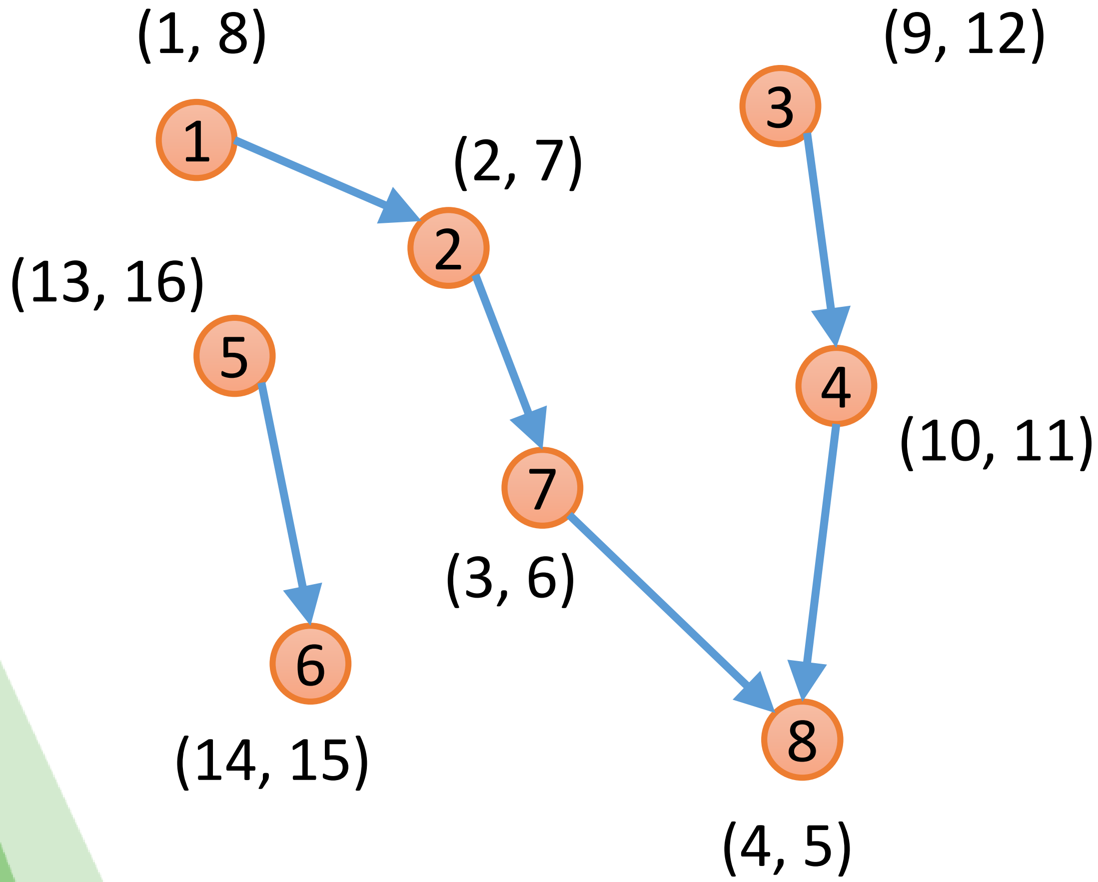


時間戳記

- 時間戳記的關鍵在「離開戳記」！
- 對於兩個點 p,q：
  1. 如果 p 可以走到 q，且 p 先被 DFS 拜訪
     - 離開 q 後還會回到 p，p 的離開戳記比 q 還要大
  2. 如果 p 可以走到 q，且 q 先被 DFS 拜訪
     - 離開 q 後才會走到 p，p 的離開戳記比 q 還要大
  3. 如果兩點可以互通
     - 誰的離開戳記比較大與 DFS 順序有關，誰先拜訪誰的離開戳記就比較大
- 我們可以透過離開戳記的大小判定輩分大小！

方法二：DFS 時間戳記

1. 對整張圖進行一次 DFS 遍歷，並在途中記錄時間戳記
2. 根據離開戳記遞減排列即形成一組解



5	6	3	4	1	2	7	8

無解判定

- 可以互通（圖上存在環）的情形下，「結束戳記大的輩分就大」不一定成立
- 必須另外判定無解情形
  - DFS 過程中順便判定
  - 給出解後驗證合法性

總結

- 空間複雜度：$O(n+m)$
- 時間複雜度：$O(n+m)$
- 常見用途：
  - DAG 判定
  - 輔助解決具有依賴關係的問題
- 時間戳記以後還有戲分，務必要好好理解喔 >.^