



Graph

by music960633

Sprout



課程內容

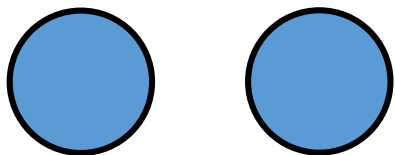
- 1. 介紹、定義
- 2. 如何存圖
- 3. 圖上DFS、BFS
- 4. 連通塊數量
- 5. 二分圖判定

Sprout



介紹、定義

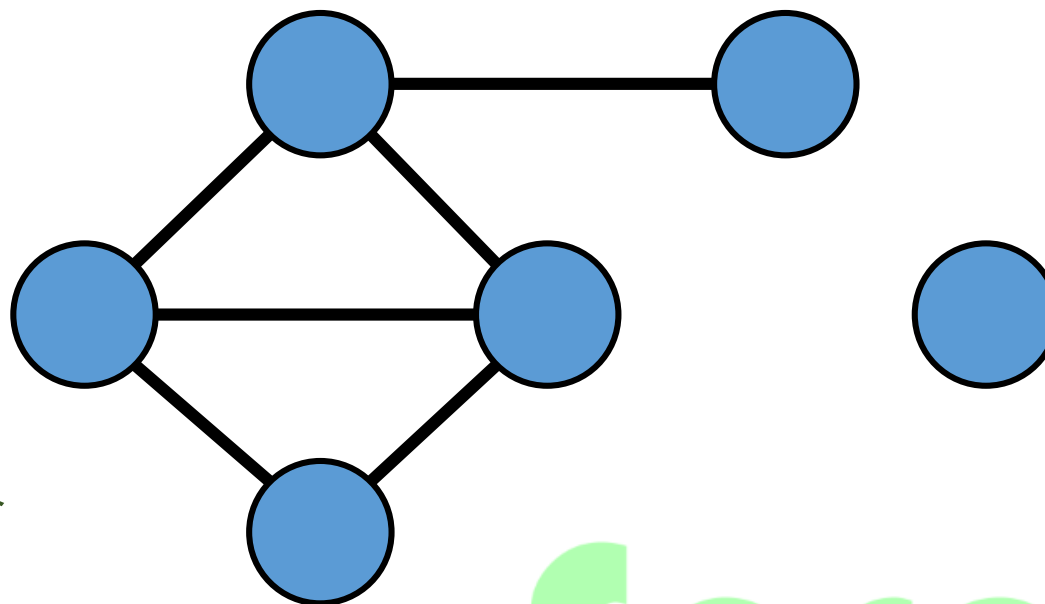
- 基本元素：點和邊
- 點(vertex)



- 邊(edge)



- 圖：點的集合+邊的集合



Sprout

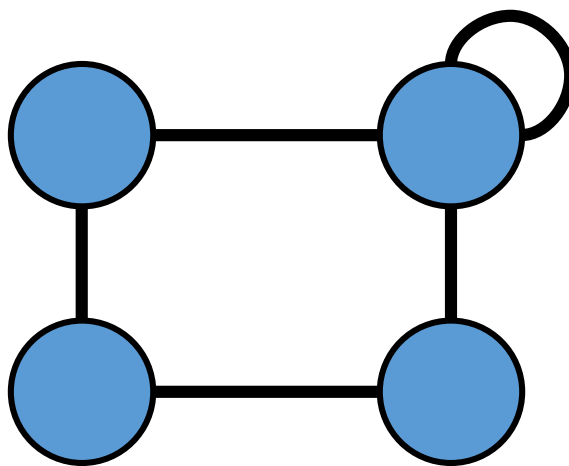
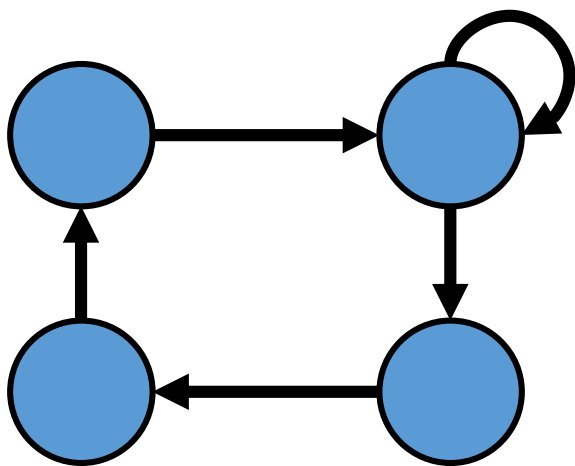


介紹、定義

- 有向邊、無向邊



- 有向圖(directed graph)、無向圖(undirected graph)

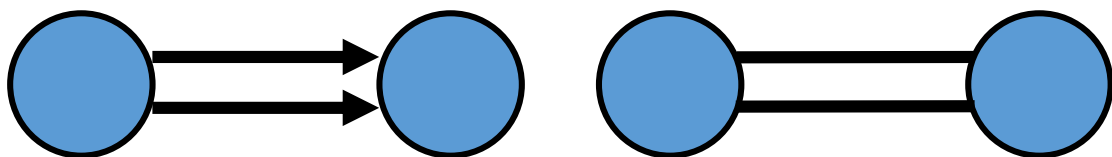


Sprout

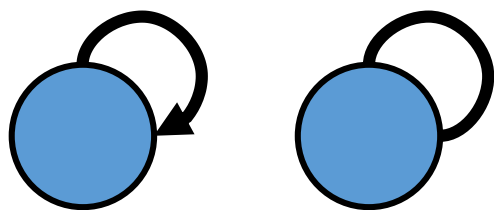


介紹、定義

- 重邊



- 自環

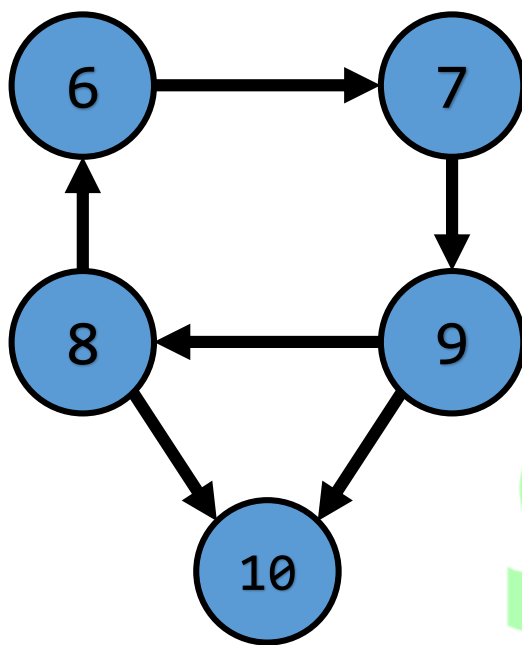
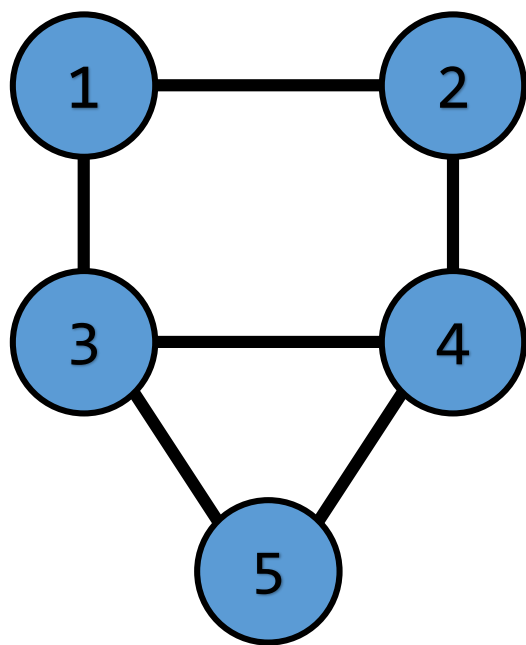


Sprout



介紹、定義

- 度(degree)：和一個點 v 有相關的邊的數量
- 入度(in-degree)：終點為 v 的邊數
- 出度(out-degree)：起點為 v 的邊數

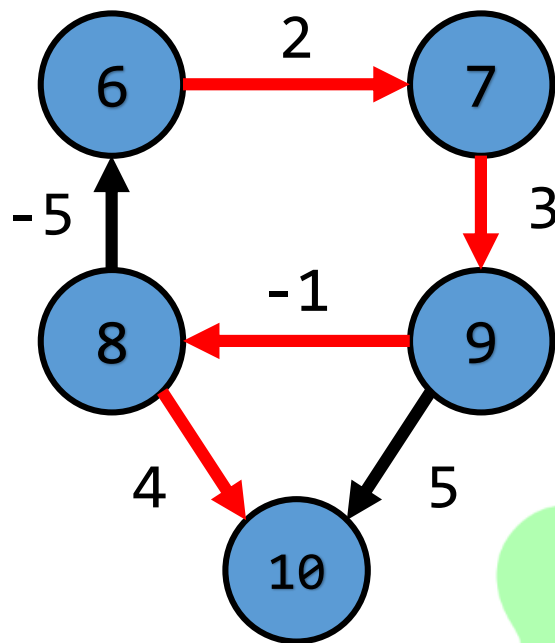
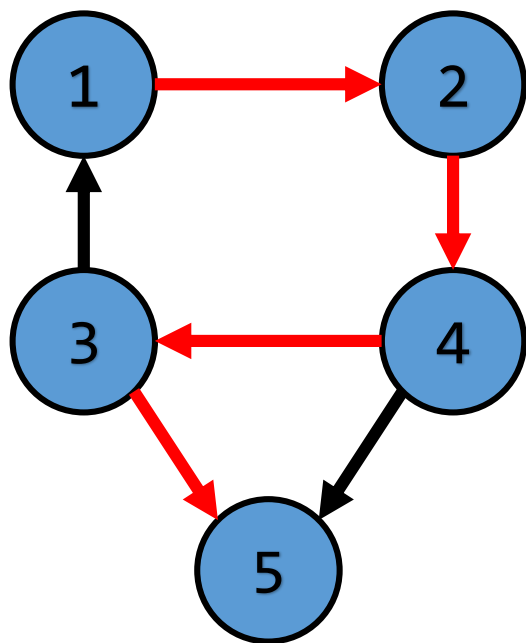


Sprout



介紹、定義

- 路徑：由頭尾相連的邊組合而成的集合
- 路徑長：路徑上邊的數量或邊的權重總合

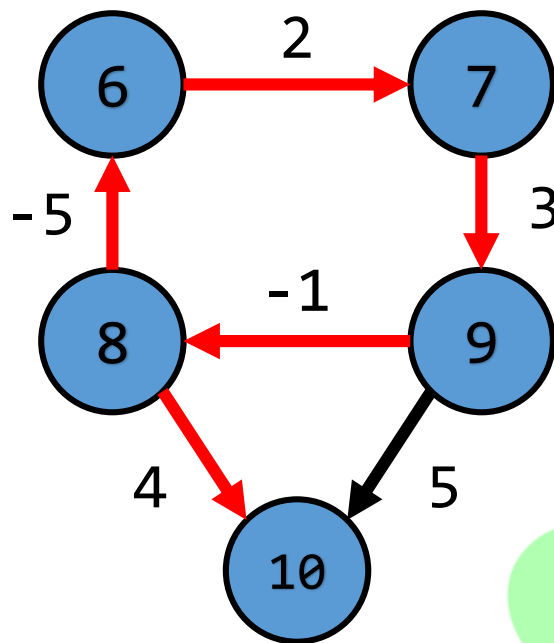
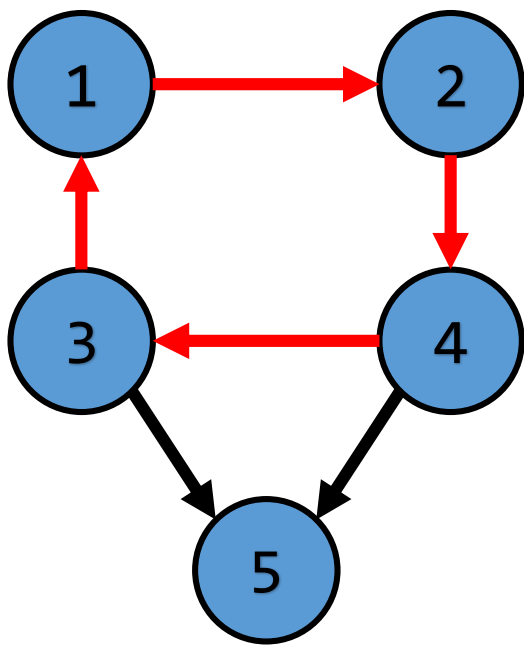


Sprout



介紹、定義

- 環：起點和終點為同一點的路徑
- 如果沒特別說明，路徑也可以包含環

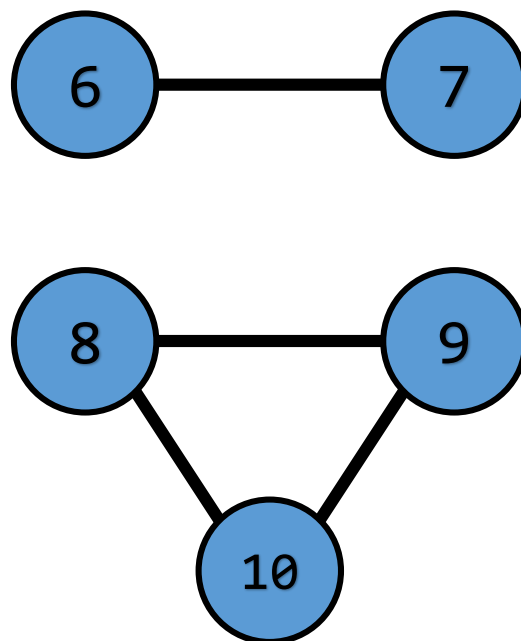
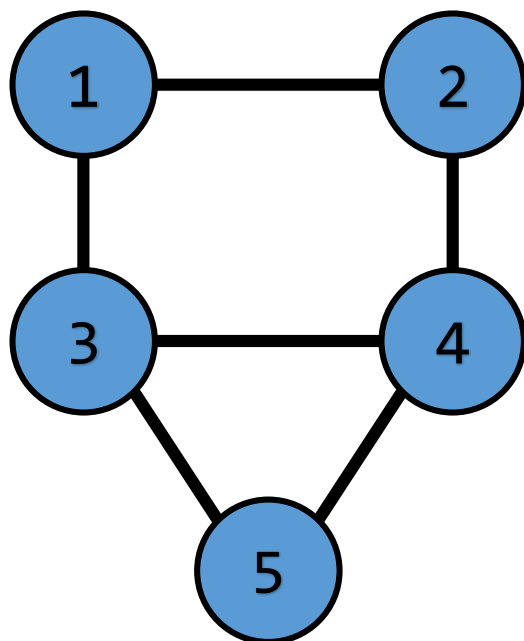


Sprout



介紹、定義

- 連通圖：圖上任相異兩點間必存在一條路徑

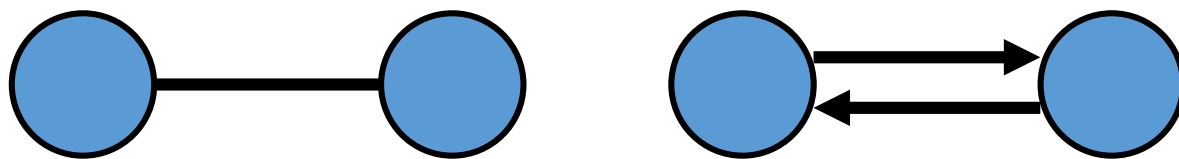


Sprout



如何存圖

- 當然可以像之前學過的linked list，將一個vertex視為一個node，並用指標記錄和該點相鄰的點
 - 缺點：實做不方便
- 無向邊可想成兩條相反的有向邊

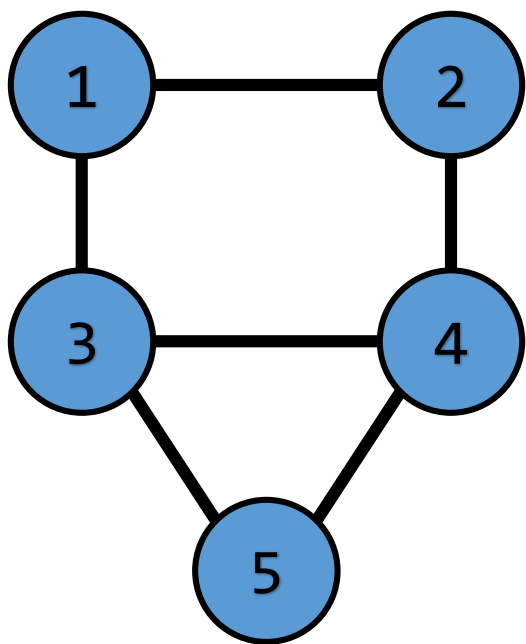


Sprout



如何存圖

- 方法一：相鄰矩陣(adjacent matrix)
- 無向圖



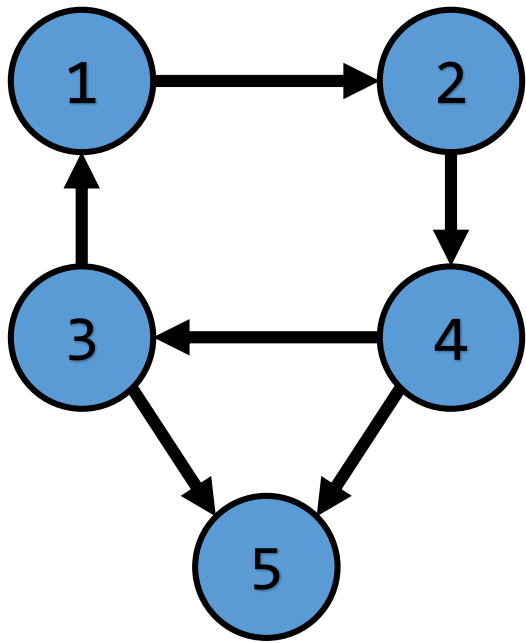
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

sprout



如何存圖

- 方法一：相鄰矩陣(adjacent matrix)
- 有向圖



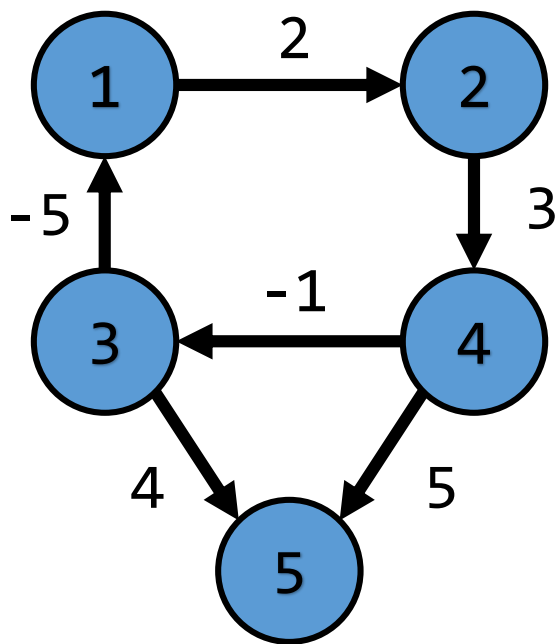
	1	2	3	4	5
1	0	1	0	0	0
2	0	0	0	1	0
3	1	0	0	0	1
4	0	0	1	0	1
5	0	0	0	0	0

aprount



如何存圖

- 方法一：相鄰矩陣(adjacent matrix)
- 有向圖+邊有權重



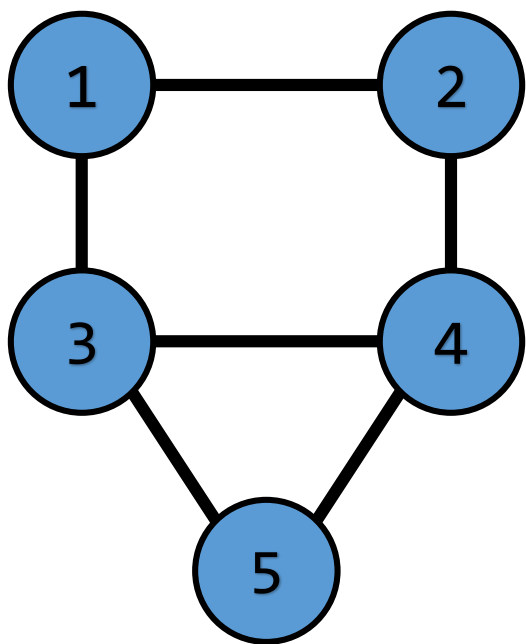
	1	2	3	4	5
1	?	2	?	?	?
2	?	?	?	3	?
3	-5	?	?	?	4
4	?	?	-1	?	5
5	?	?	?	?	?

aprount



如何存圖

- 方法二：相鄰串列(adjacent list)
- 無向圖



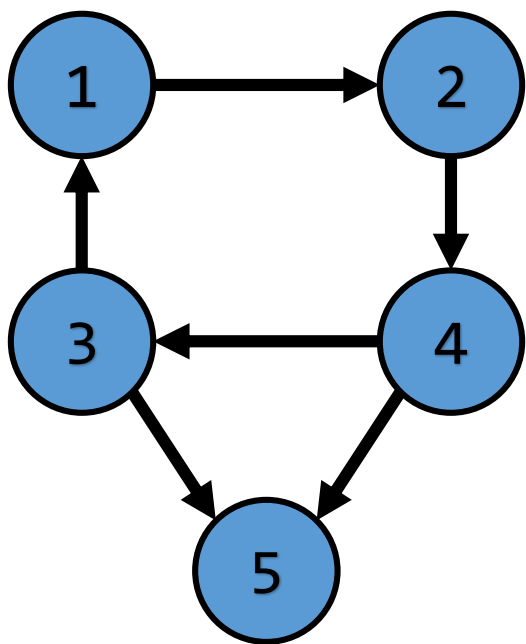
1	2	3		
2	1	4		
3	1	4	5	
4	2	3	5	
5	3	4		

Sprout



如何存圖

- 方法二：相鄰串列(adjacent list)
- 有向圖



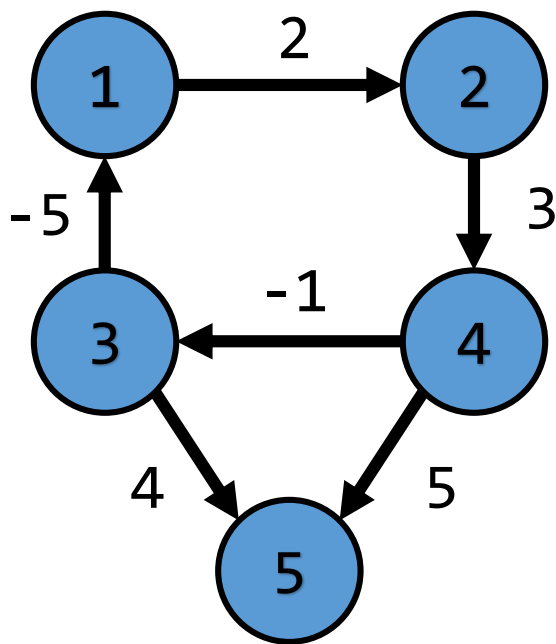
1	2			
2	4			
3	1	5		
4	3	5		
5				

Sprout



如何存圖

- 方法二：相鄰串列(adjacent list)
- 有向圖+邊有權重



1	(2,2)	
2	(4,3)	
3	(1,-5)	(5,4)
4	(3,-1)	(5,5)
5		

Sprout



如何存圖

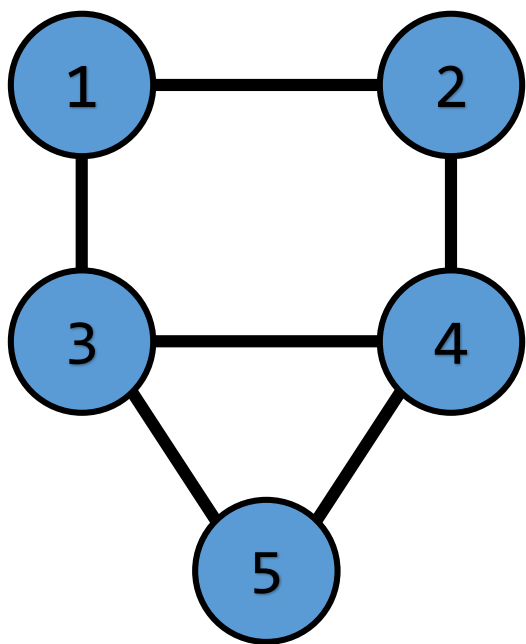
- 比較
- 相鄰矩陣
 - 空間複雜度： $O(V^2)$
 - 查詢兩個點之間是否有邊： $O(1)$
 - 遍歷一個點 v 周圍的邊： $O(V)$
- 相鄰串列
 - 空間複雜度： $O(E)$
 - 查詢兩個點之間是否有邊： $O(\text{degree}(v))$
 - 遍歷一個點 v 周圍的邊： $O(\text{degree}(v))$

Sprout



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



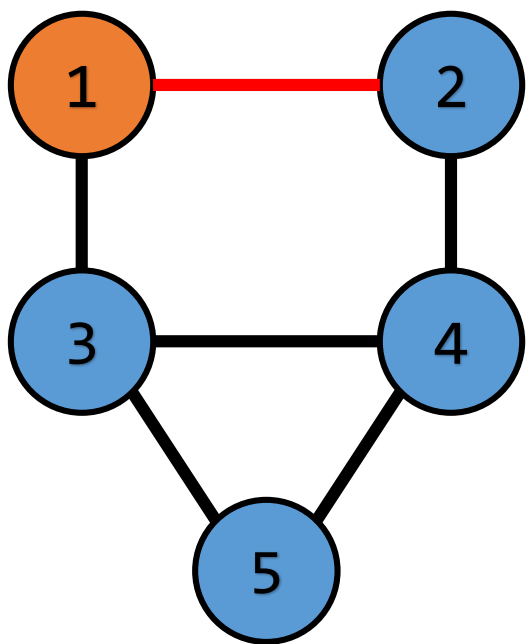
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

sprout



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



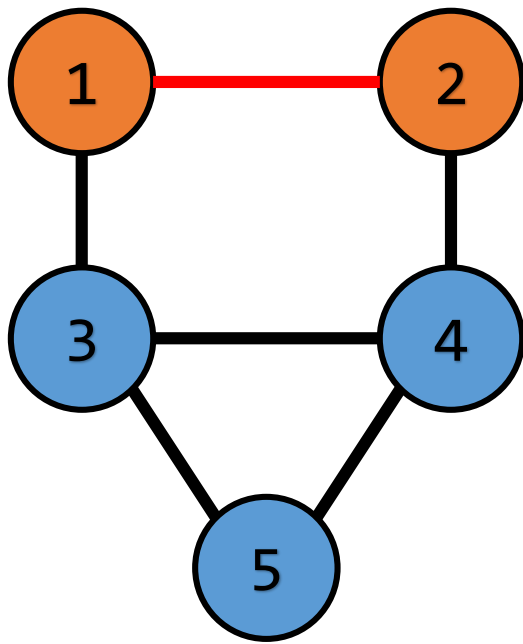
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

sprout



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



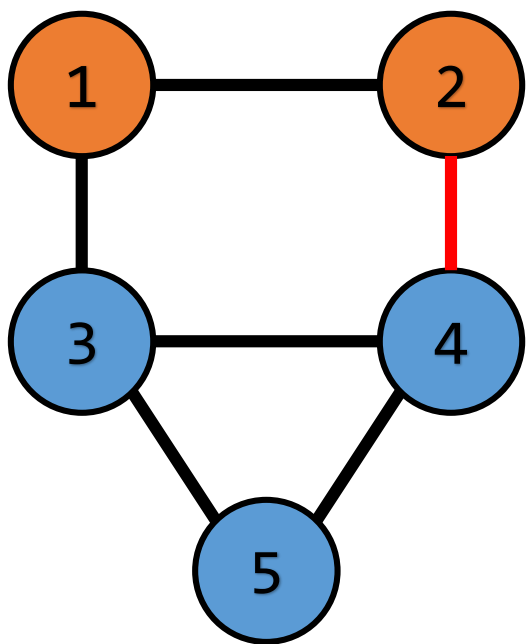
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

sprout



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



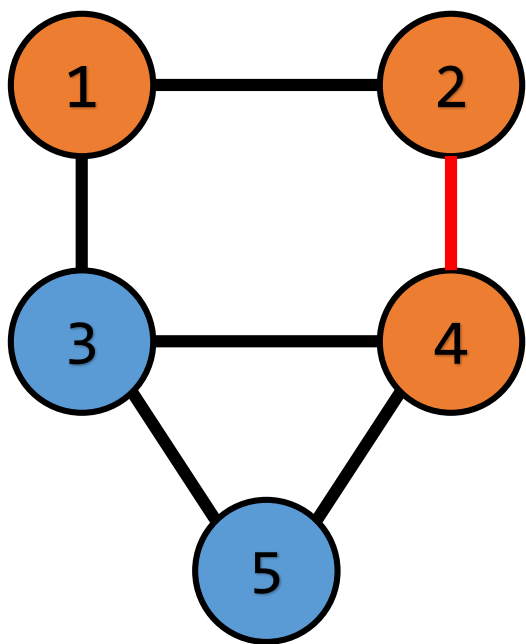
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

sprout



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



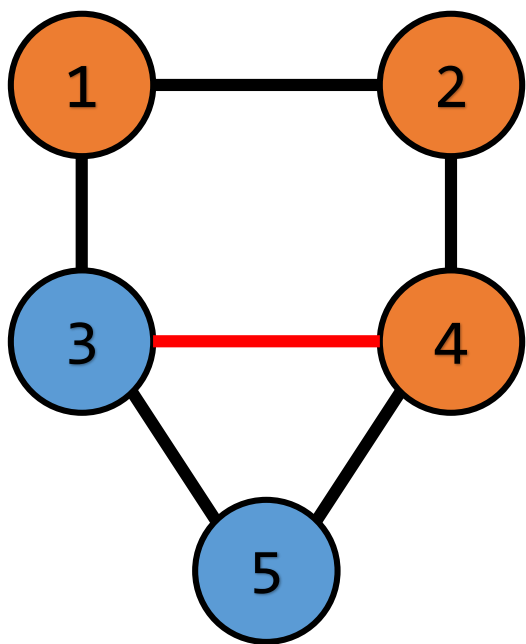
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

aproot



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS :



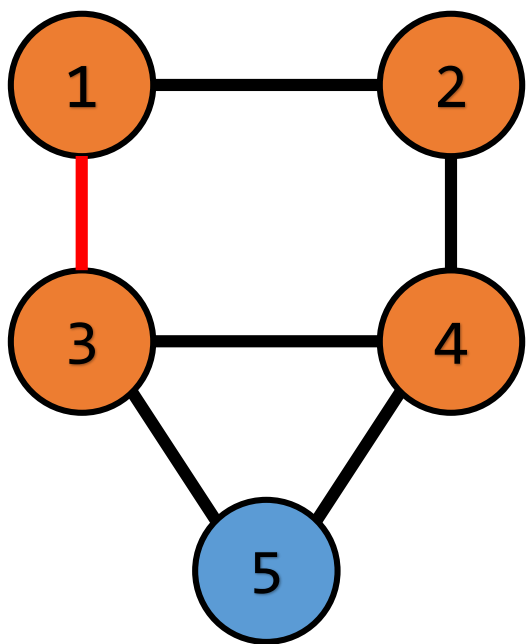
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

sprout



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



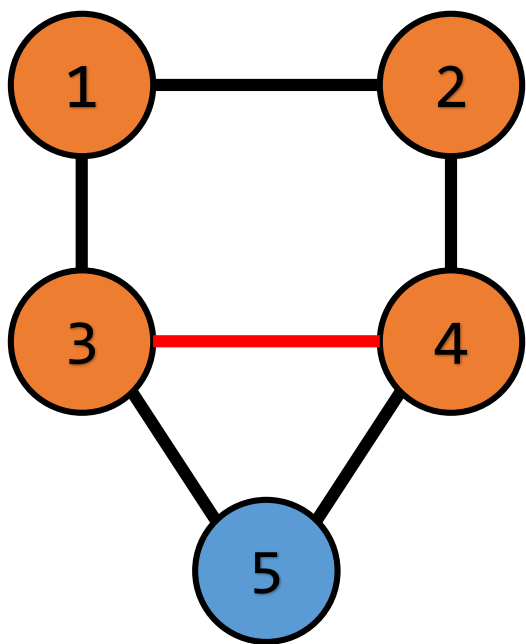
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

sprout



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS :



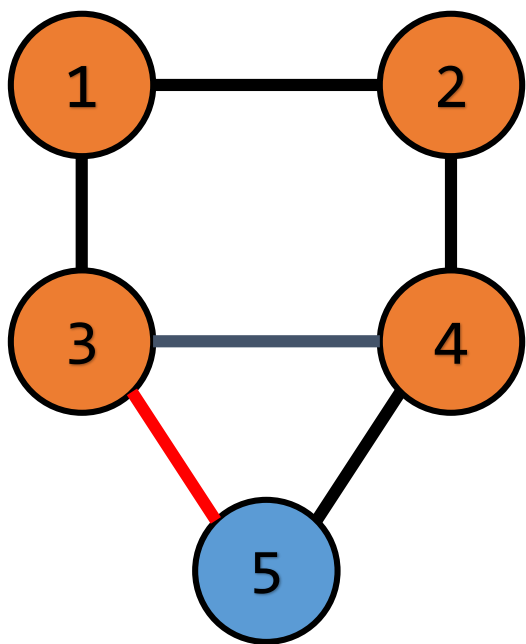
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

sprout



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



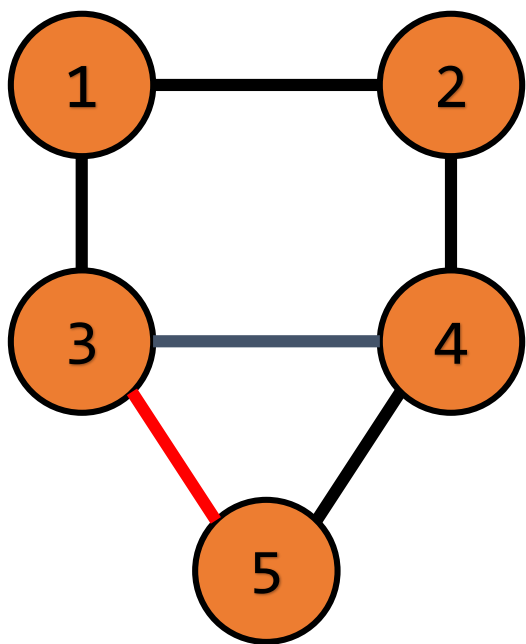
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

aprouk



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



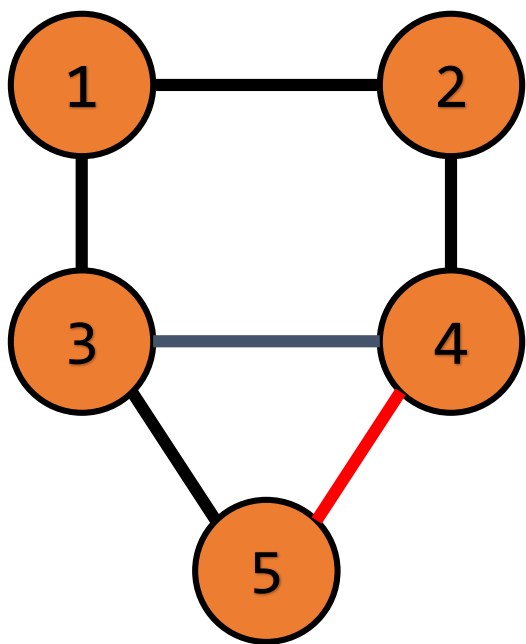
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

aprount



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



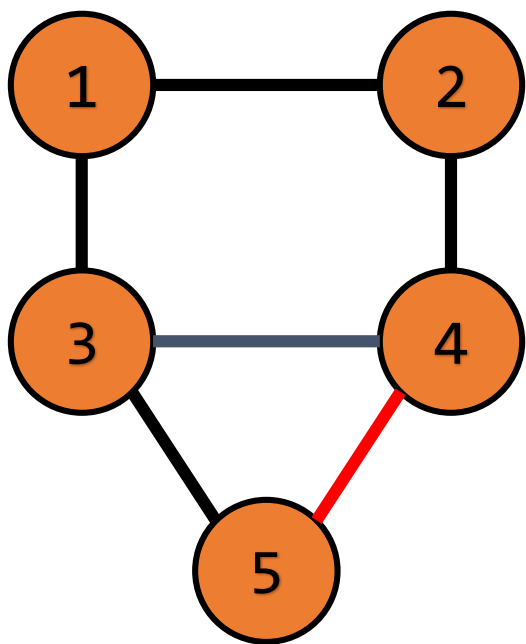
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

aprount



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



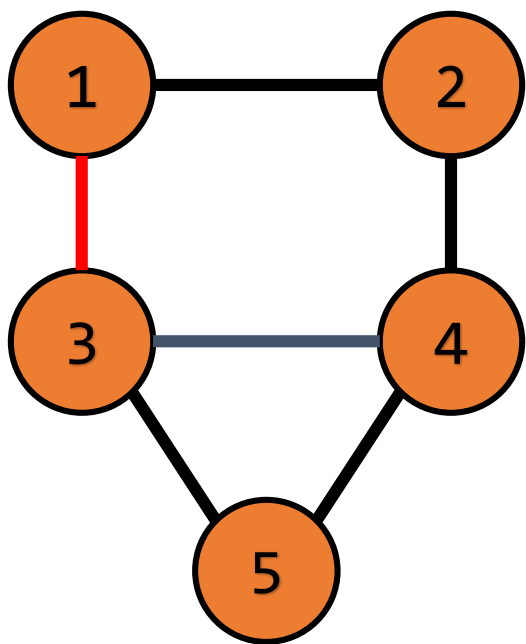
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

aprount



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



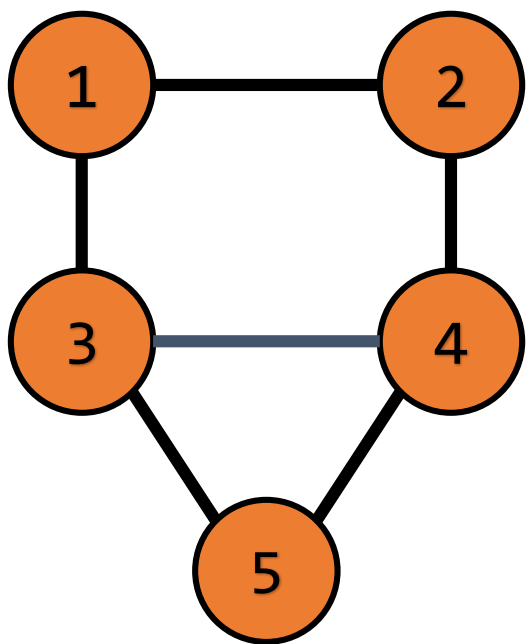
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

aprount



圖上DFS、BFS

- 如果你對tree的DFS、BFS了解了話應該很快可以上手
- DFS：



	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1
5	0	0	1	1	0

aprount



圖上DFS、BFS

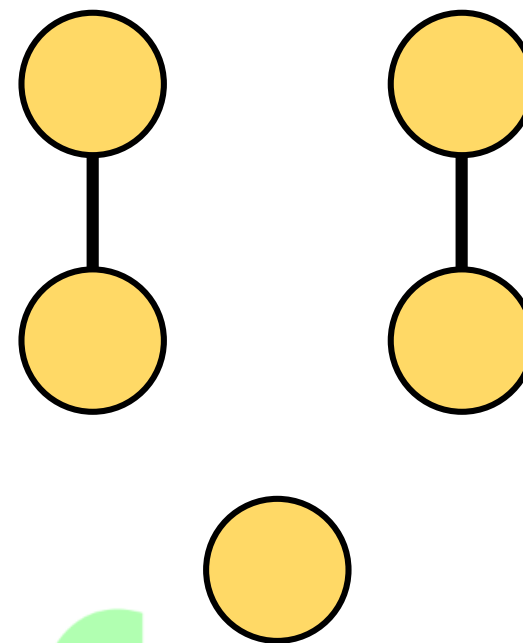
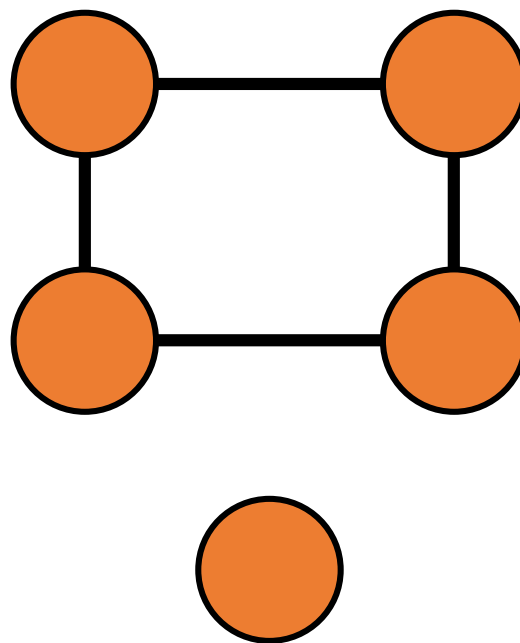
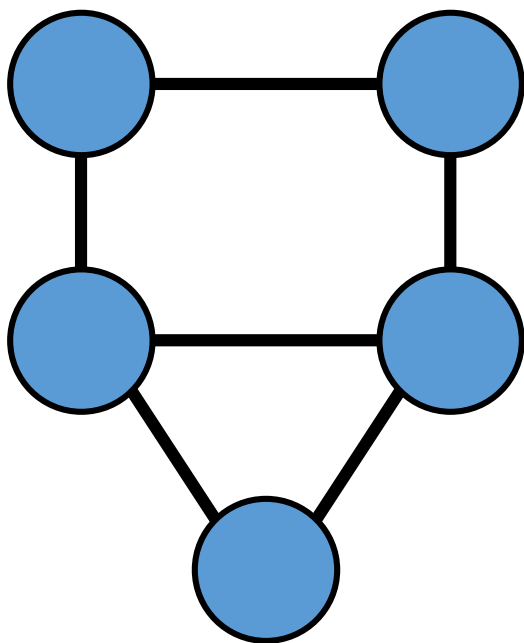
- DFS、BFS注意事項
 - 1. 要記錄一個點是否被拜訪過，以免重複拜訪
 - 2. 走訪前記得判斷是否拜訪過
 - 3. 走訪後記得將該點設為已經拜訪過
- 很多RE、TLE就是沒注意這些而造成的！

Sprout



連通塊數量

- 如何判定一張無向圖中有幾個連通塊呢？



Sprout



連通塊數量

- 注意到：做完一次DFS後，所有在同一個連通塊中的點都會被拜訪到，不在同一個連通塊中就不會被拜訪到
- 計算做了幾次DFS後全部的點都被拜訪到就好了！

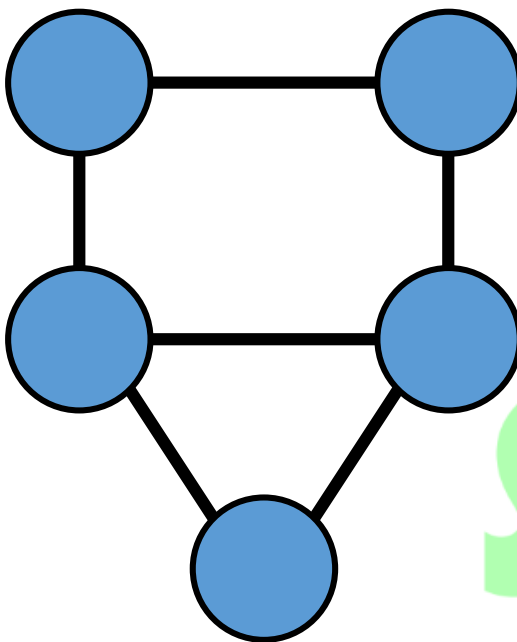
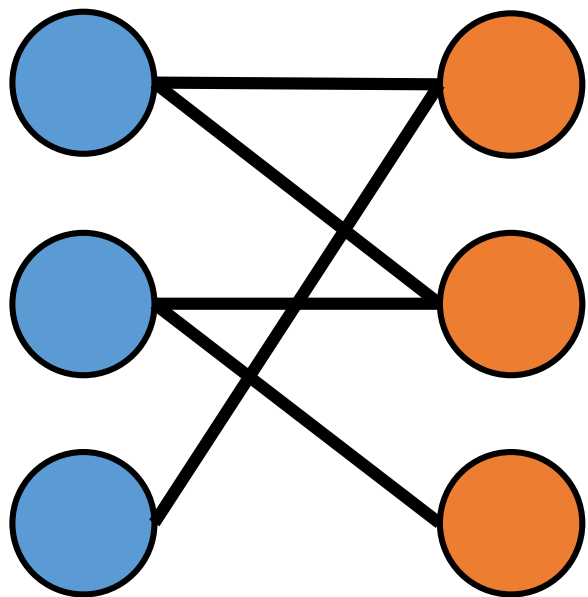
```
2  int count = 0;
3
4  for(int i=1; i<=V; ++i)
5      visit[i] = false;
6
7  for(int i=1; i<=V; ++i){
8      if(!visit[i]){
9          dfs(i);
10         ++count;
11     }
12 }
```

Sprout



二分圖判定

- 二分圖：一個無向圖中的點可以被分成兩個集合，使得同集合內的點互不相鄰
- 黑白染色：將圖上的點塗成黑或白，使得每條邊的兩端必不同色

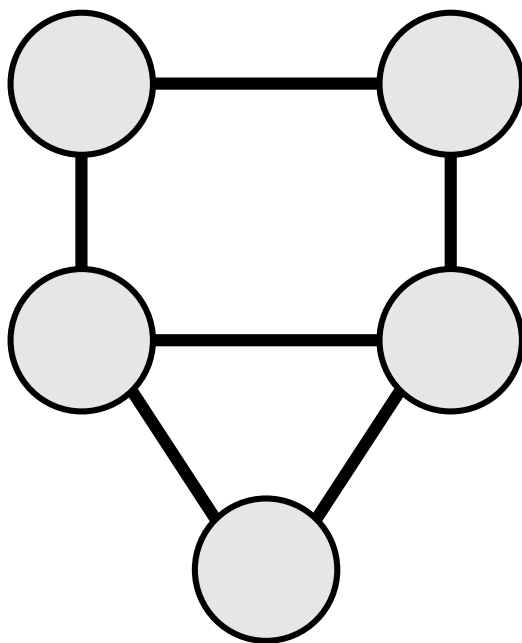


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖

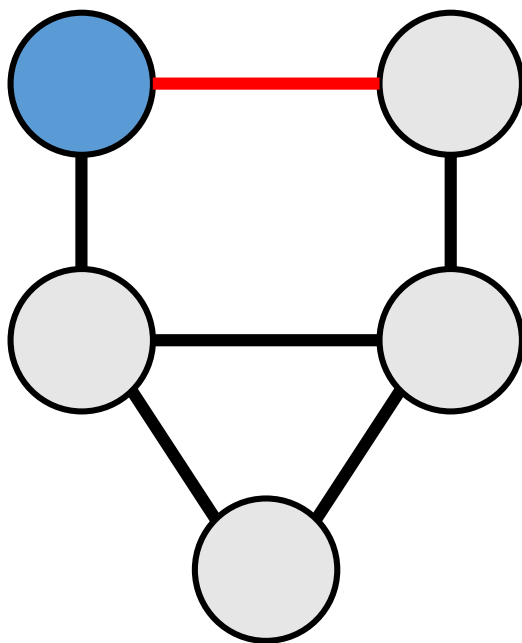


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖

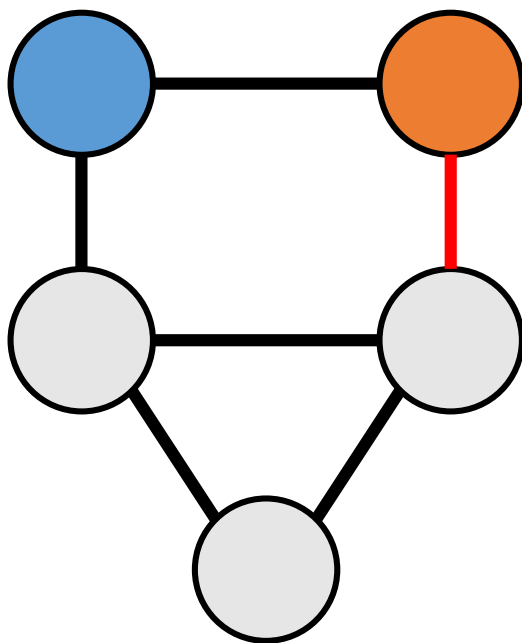


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖

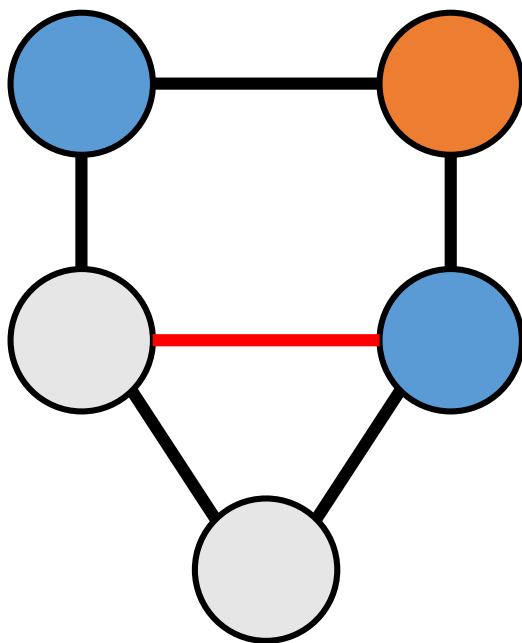


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖

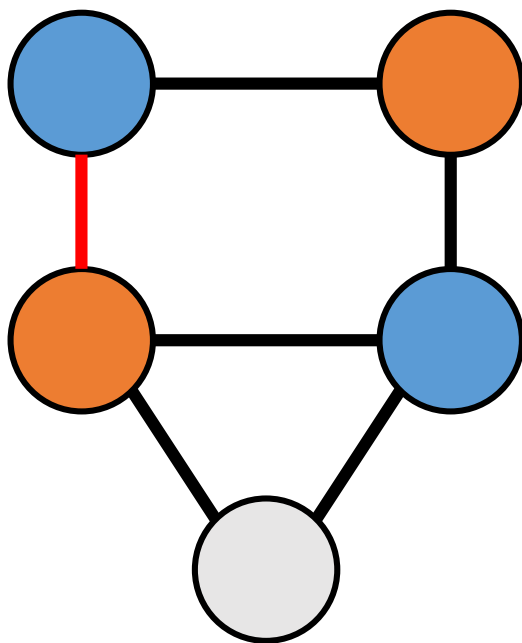


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖

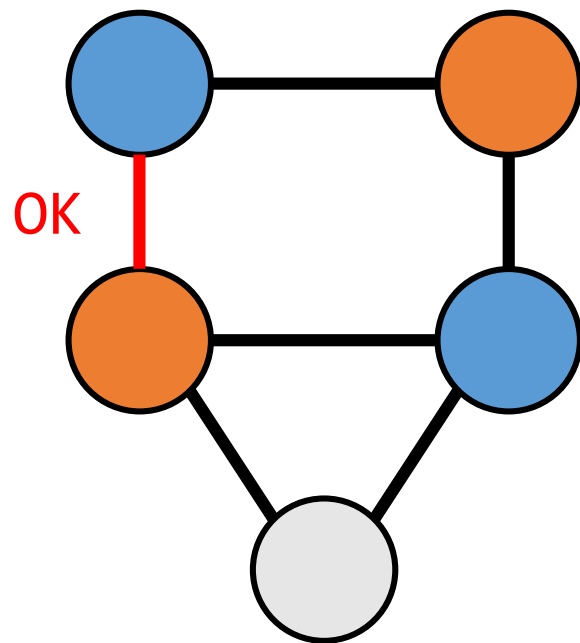


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖

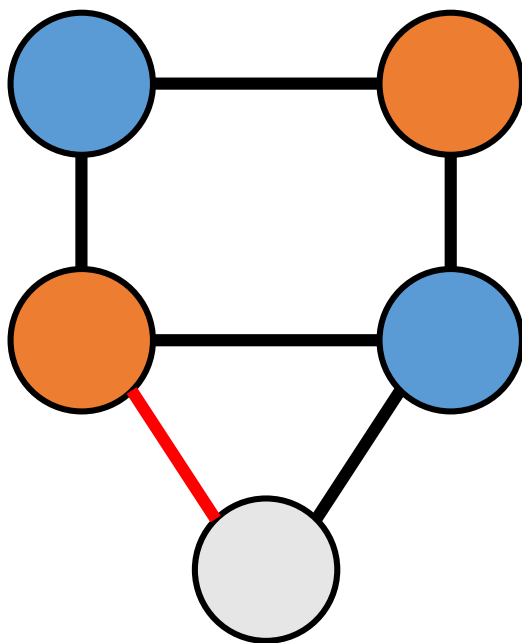


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖

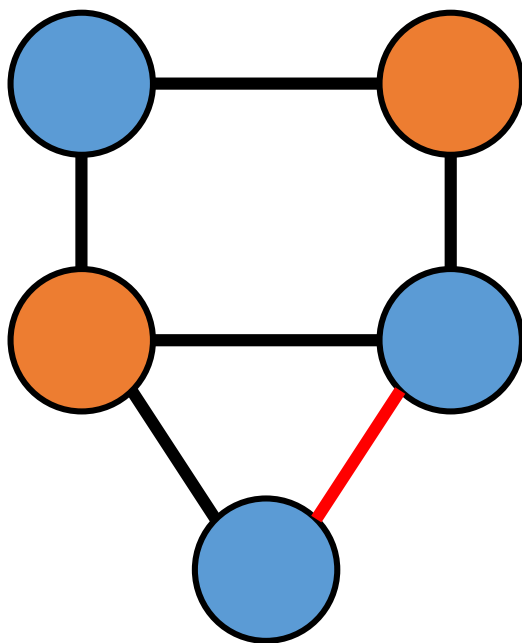


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖

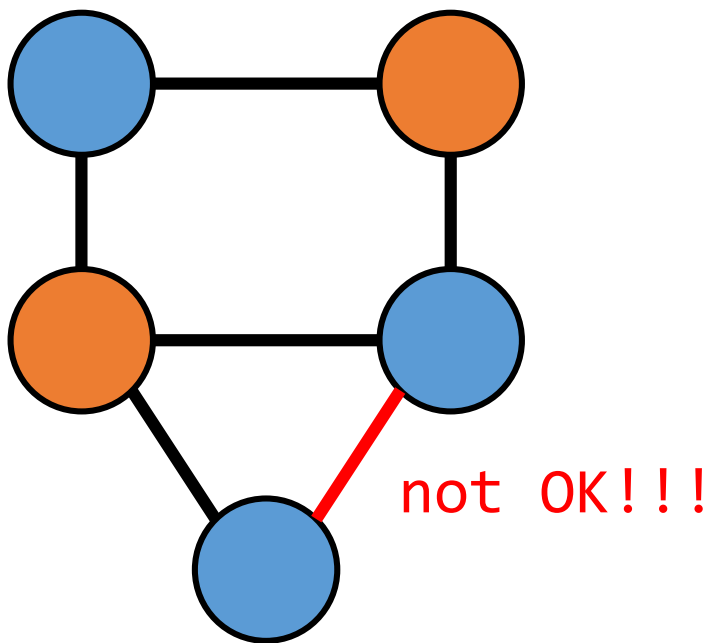


Sprout



二分圖判定

- 如何判定一個無向圖是否為二分圖呢？
- 1. 隨便找一個點，塗上黑或白
- 2. 從這一點DFS、BFS，並將相鄰的點塗上相異的顏色
- 3. 如果塗色過程發生矛盾，則不是二分圖，否則就是二分圖



Sprout



結語

- 圖論很重要，很多看起來跟圖論沒關係的問題其實可以轉成圖論來解，因此基本的DFS、BFS一定要熟
- 很多圖上的問題都可以只靠DFS、BFS做出來
- 大家加油！

Sprout