演算法	時間	sparse graph $\lvert E\rvert=\theta(\lvert V\rvert)$	dense graph $\lvert E\rvert=\theta(\lvert V\rvert^2)$
Kruskal	$O(\lvert E\rvert\log\lvert V\rvert)$	$O(\lvert V\rvert\log\lvert V\rvert)$ 較適合	$O(\lvert V\rvert^2\log\lvert V\rvert)$
Prim	$O(\lvert V\rvert^2)$	$O(\lvert V\rvert^2)$	$O(\lvert V\rvert^2)$ 較適合

精選例題 40

關於無向圖形 G 的敘述，何者可以正確的定義 tree？

(A) G 是 connected 而且是 acyclic。

(B) G 是 connected 而且恰好是 n-1 個 edges。

(C) G 是 acyclic 而且恰好是 n-1 個 edges。

(D) G 是 connected，而且刪除任何一個 edge，都會使得圖形 disconnects。

(E) G 非 complete graph，而且再加上任一個 edge，都會使得圖形恰好產生一個迴路，此迴路包含新加入的邊。

解答：(A) (B) (C) (D) (E)

8-6 雙連通單元(Biconnected Components)

✧ 要點：關節點(Articulation Point)定義

若且唯若 v 為圖形 G 的關節點，則在圖形 G 中，若刪除頂點 v(以及其 incident edges)之後，會使 G 分解成 2 個(或以上)的不連通單元。

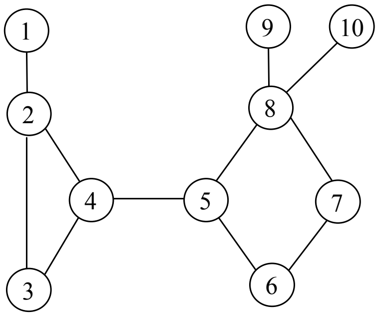
✧ 要點：雙連通圖形(Biconnected Graph)定義

若且唯若圖形 G 為雙連通圖形，則在圖形 G 中，不包含任何關節點。

✧ 要點：雙連通單元(Biconnected Component)定義

雙連通單元為圖形中的最大雙連通子圖。意即，包含雙連通單元的更大子圖（母集合），皆非雙連通子圖。

➢ **範例：**下圖中頂點 2、4、5、8 爲節點。



✧ 要點：雙連通單元的求法

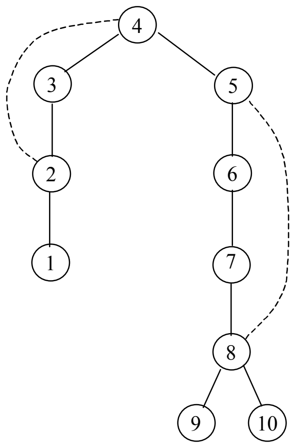
1. 建立圖形的 DFS spanning tree，並計算出每一個頂點 u 的 dfn[u] 值。
   (1) dfn[u](depth first number of v)是代表頂點 u 在 DFS 追蹤時，是第幾個追到的頂點。因此，若 u 比 v 先追蹤(u 是 v 的 ancestor)，則 dfn[u] < dfn[v]。
   (2) 在 DFS spanning tree 中，若頂點 u 是 v 的祖先，則 dfn[u] < dfn[v]，而 root 的 dfn 值爲 1。
2. 在進行 DFS 追蹤時，可以將edges 分類不同的類型
   (1) tree edges：DFS spanning tree 中有使用到的 edges。
   (2) non-tree edges：DFS spanning tree中沒有使用到的 edges。又分爲 backward edges、forward edges 及 cross edges 三種。
      ① backward edge：若 non-tree edge <u,v> 中的 v 爲 u 的祖先，則 <u,v> 是 backward edge。
      ② forward edge：若 non-tree edge <u,v> 中的 v 爲 u 的後代(注意：因爲 <u,v> 爲 non-tree edge，故 u 並非 v 的父親)，則 <u,v> 是 forward edge。
      ③ cross edge：若 non-tree edge <u,v> 中的 v 與 u 互相之間並沒有祖先後代的關係，則 <u,v> 爲 cross edge。
3. edges 分類的重要特性

<u,v>的分類	backward edge	forward edge	cross edge
DFS中，由u沿<u,v>追蹤v時	(1)v 為 being_visited (2)dfn[u]>dfn[v]	(1)v 為 finish (2)dfn[u]<dfn[v]	(1)v 為 finish (2)dfn[u]>dfn[v]
無向圖形	=forward edge	=backward edge	沒有 cross edge
有向圖形	≠forward edge	≠backward edge	有 cross edge

4. 計算各頂點的low[v] 值：low[v] 代表以 v 為 root 的子樹，沿 backward edge 所能到達的祖先中，最小的dfn值。

$$\text{low}[v]=\min\{\ \text{dfn}[v],\ \min\{\text{low}[w] \mid w \text{ 是 } v \text{ 的 child}\},\ \min\{\text{dfn}[w] \mid (v,w) \text{ 是 back edge}\}\ \}$$

➢ **範例：**以頂點 4 為起點，建立 DFS Spanning Tree，並計算 dfn 與 low。



頂點	dfn	low
1	4	4
2	3	1
3	2	1
4	1	1
5	5	5
6	6	5
7	7	5
8	8	5
9	9	9
10	10	9

5. 判斷頂點 u 是否為關節點
   (1) 頂點 u 為 DFS spanning tree 的 root 時，若頂點 u 有兩個(或以上)的 children。
   (2) 頂點 u 不是 DFS spanning tree 的 root 時，則頂點 u 至少存在一個 child v， 滿足 low(v)≧dfn(u) 時，頂點 u 即為關節點。意即，以 v 為 root 的子樹，除了經由 u 之外，沒有其他路徑可以到達 u 的祖先。