

# 樹狀結構的表示法

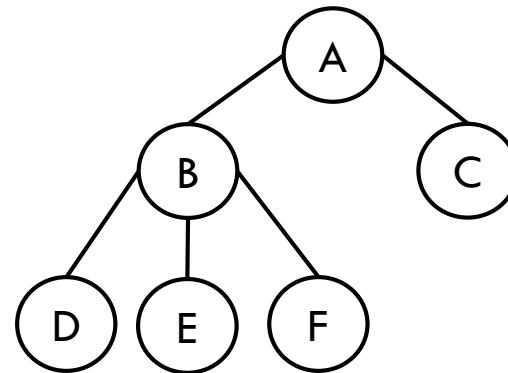
資料結構  
鍾宜玲

# 串列表示法

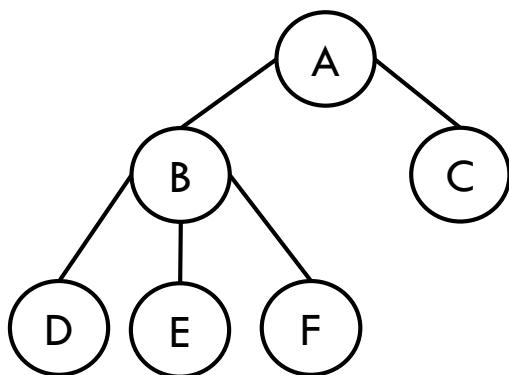


樹狀結構可以用串列來表示，則下圖為：

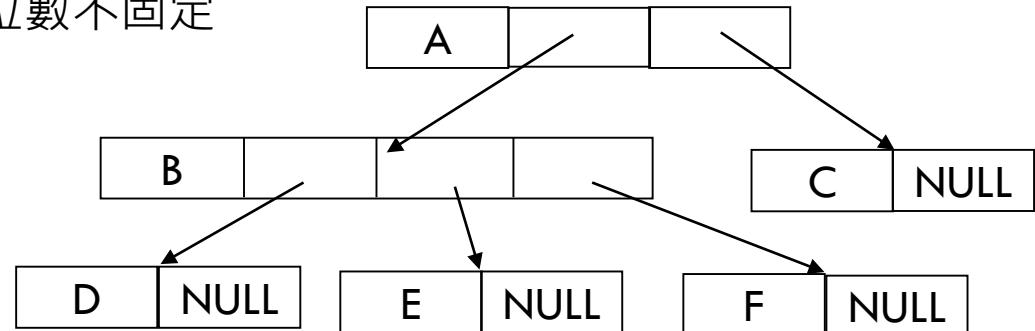
(A (B (D, E, F), C))



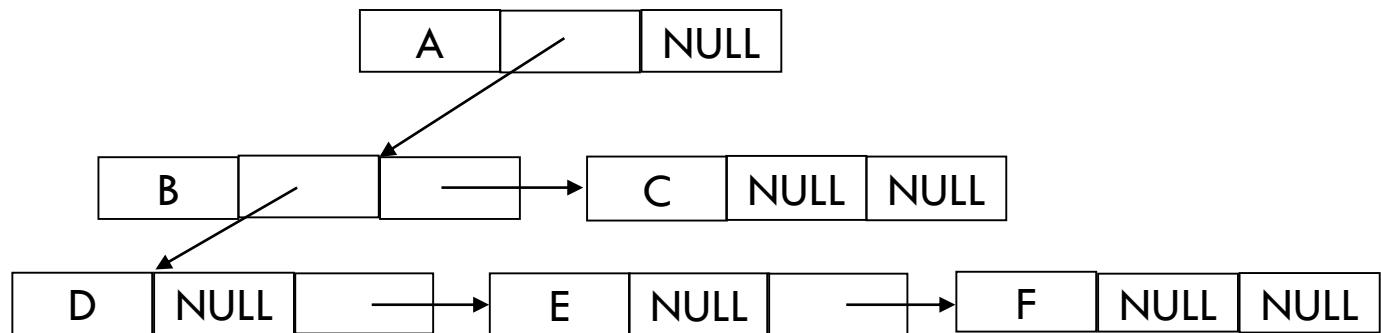
# 鏈結串列表示法



(a)節點欄位數不固定



(b)節點欄位數固定

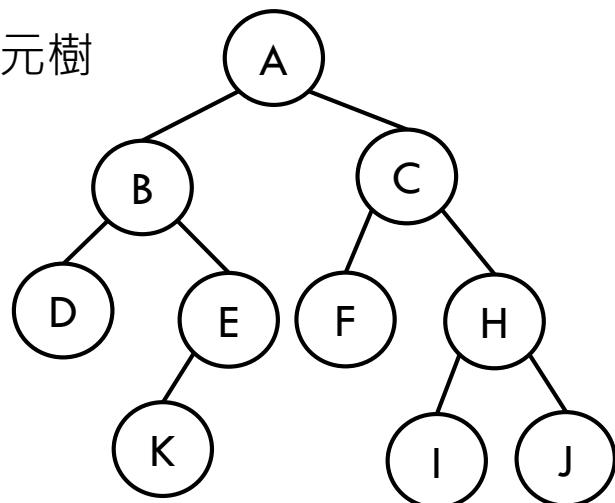


# 二元樹 (BINARY TREE)

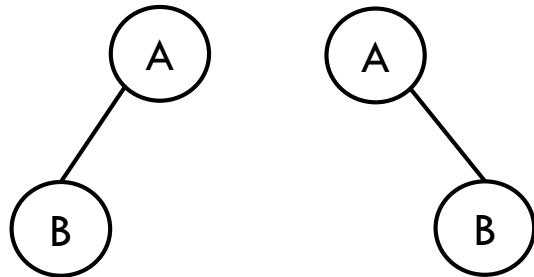


- 空樹或由樹根 (root)、左子樹 (left subtree) 或右子樹 (right subtree) 組成，其左、右子樹也是二元樹。
- 二元樹中任一節點的分支度不會超過2，且二元樹的節點個數可以是0，但一般樹不可以。

(a) 二元樹



(b) 含兩節點的兩棵不同的二元樹

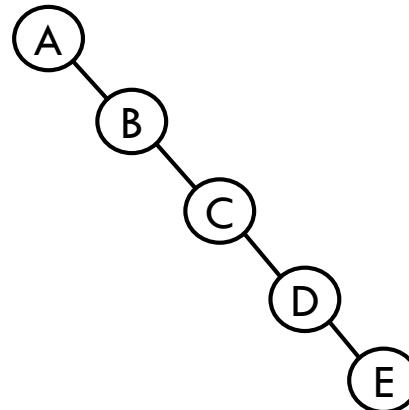
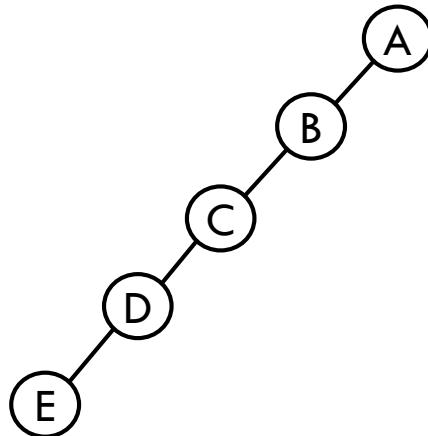


# 特殊二元樹



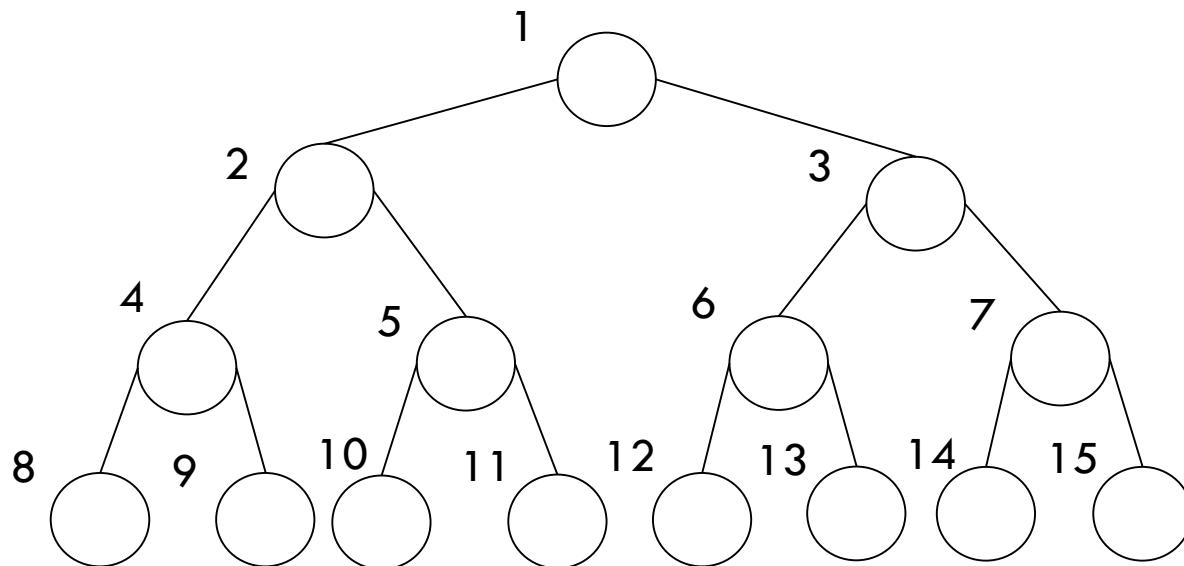
## 斜曲樹 (skewed tree)

- 二元樹其節點皆向左斜曲或向右斜曲，稱為左斜曲樹或右斜曲樹。



# 完全二元樹 (FULL BINARY TREE)

高深度k的二元樹，具有  $2^k - 1$  個節點個數

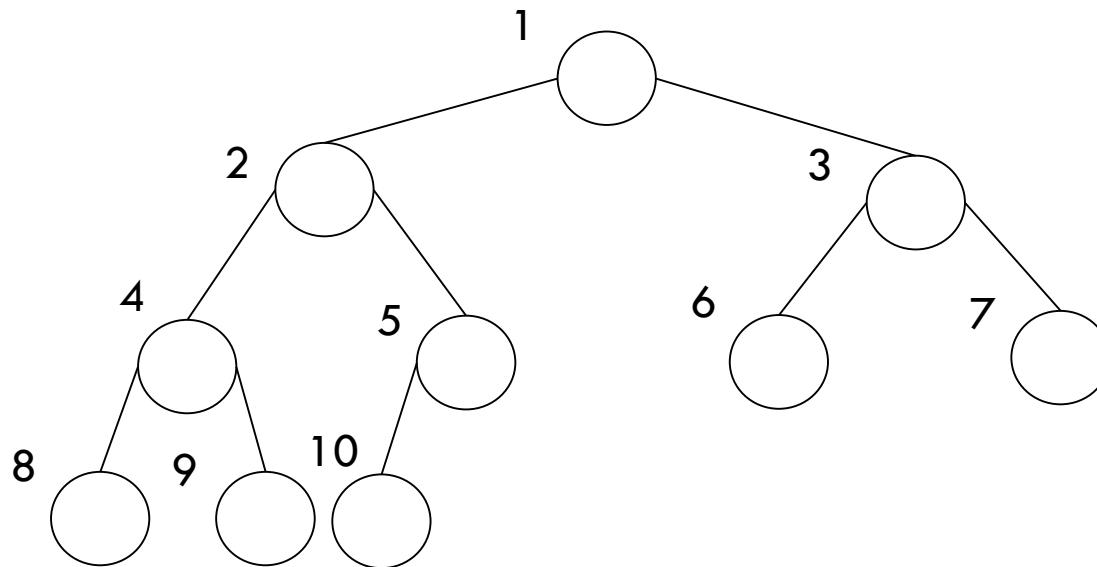


高度為4的完全二元樹

# 完整二元樹 (COMPLETE BINARY TREE)



$n$ 個節點高度為 $k$ 的二元樹，若與高度 $k$ 的完全二元樹其前 $n$ 個節點編號一一對應，則為完整二元樹 (complete binary tree)。

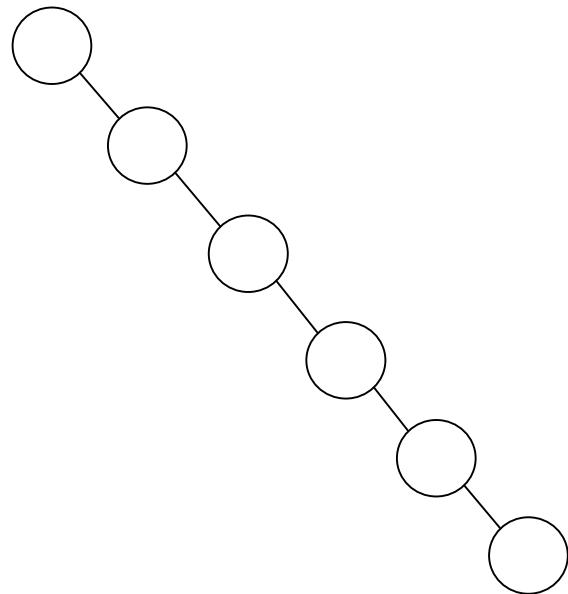


10 個節點的完整二元樹

例

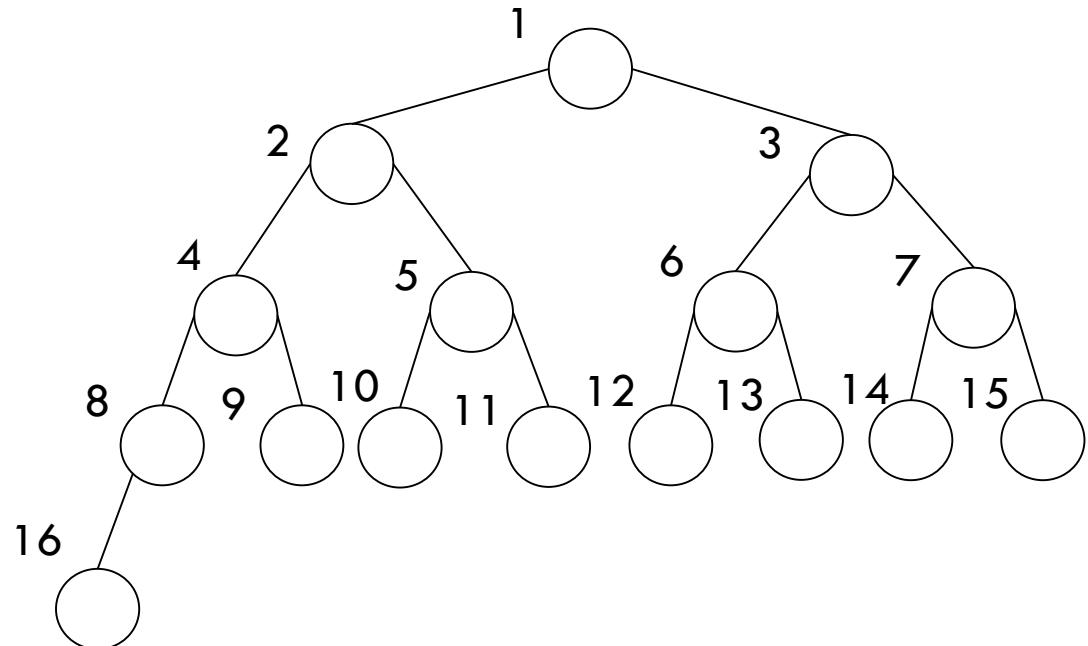


(1)

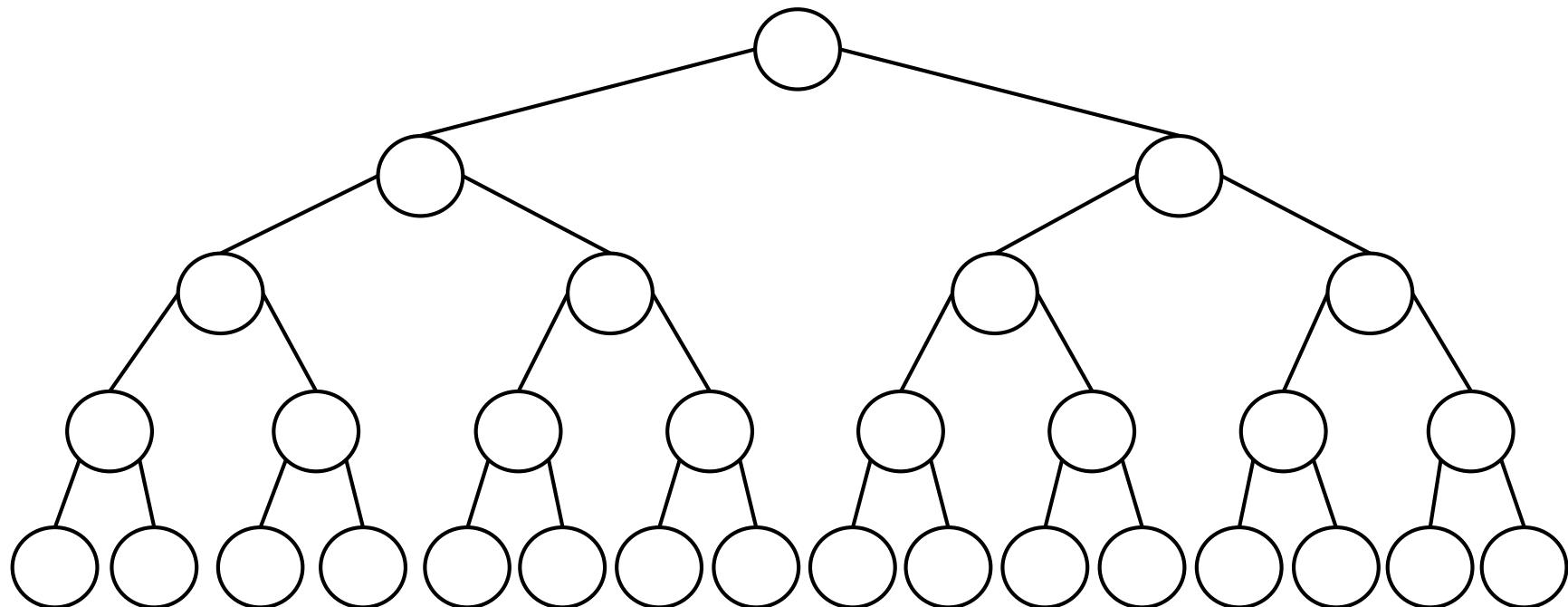


畫出高度為6的右斜曲樹

(2)高度為5的完整二元樹最少有16個節點。



(3) 高度為5的完整二元樹最多有31個節點。



# 二元樹的表示法

## 陣列表示法



樹根儲存於陣列索引值1的位置

對於編號  $i$  的節點  $p$

- 若左兒子存在，則左兒子節點的編號為  $2i$ 。
- 若右兒子存在，則右兒子節點的編號為  $2i + 1$ 。
- 若節點  $p$  不為樹根節點，則節點  $p$  的父親節點編號為  $[i / 2]$  ( 即  $i$  除以 2 取整數的商 )。

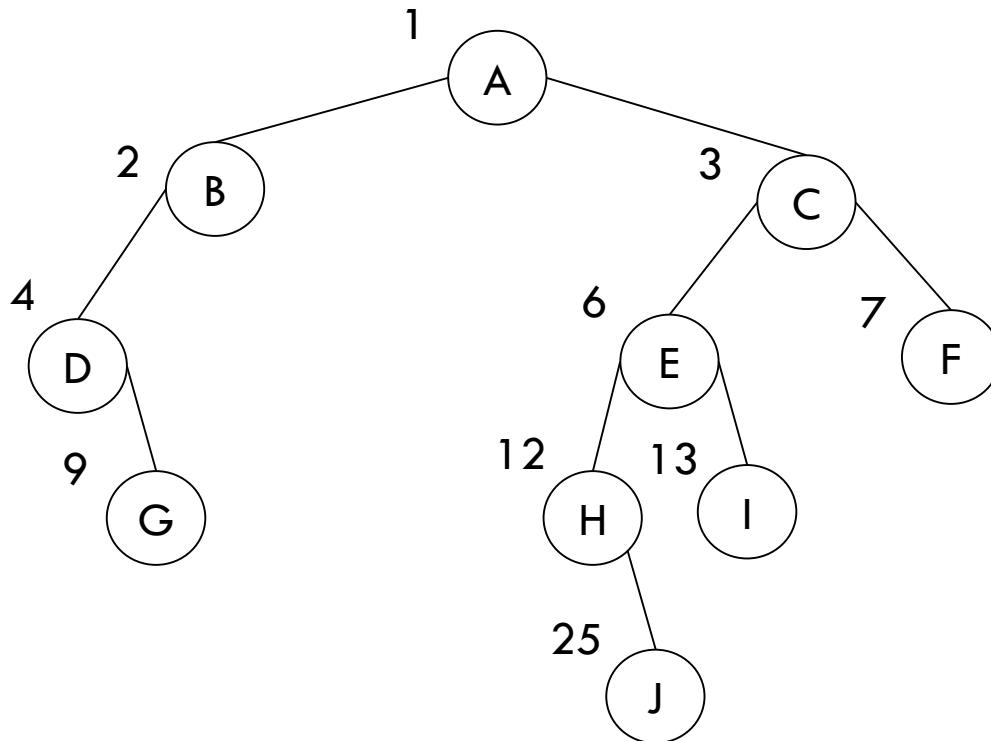
# 二元樹的陣列表示法



編號4的節點D的右兒子編號為  $4 * 2 + 1 = 9$

編號6的節點E為左兒子編號為12

編號25的節點J其父親節點編號為  $[25 / 2] = 12$ 。

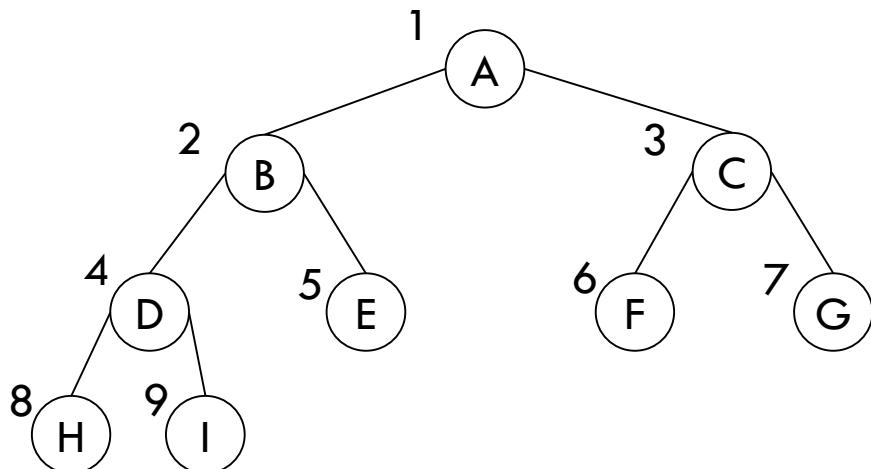


[1]	A
[2]	B
[3]	C
[4]	D
[5]	
[6]	E
[7]	F
[8]	
[9]	G
:	:
[12]	H
[13]	I
:	:
[25]	J

# 二元樹的陣列表示法

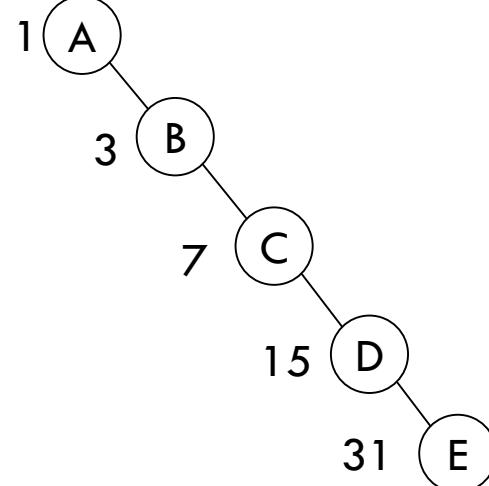


- 陣列儲存完整二元樹時最節省記憶體空間，儲存斜曲樹時最浪費記憶體空間。
- 若要插入或刪除資料需要大量移動資料。



9個節點的完整二元樹

[1]	A
[2]	B
[3]	C
[4]	D
[5]	E
[6]	F
[7]	G
[8]	H
[9]	I



高度為5的右斜曲樹

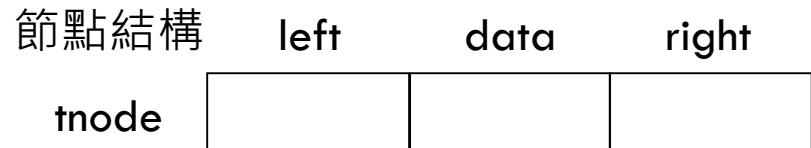
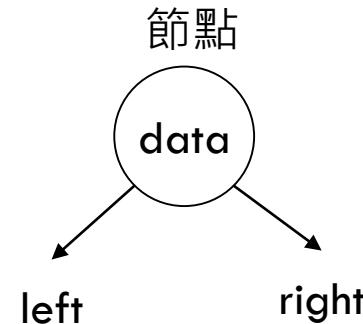
[1]	A
[2]	
[3]	B
[4]	:
[5]	C
[6]	:
[7]	D
[8]	:
[9]	E
[10]	
[11]	
[12]	
[13]	
[14]	
[15]	
[16]	
[17]	
[18]	
[19]	
[20]	
[21]	
[22]	
[23]	
[24]	
[25]	
[26]	
[27]	
[28]	
[29]	
[30]	
[31]	E

# 鏈結表示法



節點有資料欄`data`及兩個指標`left`、`right`，則節點結構定義如下：

```
typedef struct tnode{  
    char data;  
    struct tnode *left;  
    struct tnode *right;  
} TNODE;
```



# 二元樹的鏈結表示法

