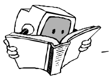
3. 後序表示法（postfix）：又稱後置波蘭表示法（reverse polish notation）。
[運算元1][運算元2][運算子]，例如：ab+。

四、表示法轉換

我們如何將中序表示法轉換成前序表示法及後序表示法呢？

1. 方法一：考慮運算子的優先權及結合性將他們括號起來，再將運算子取代相對應的括號，取代時運算子置左爲前序表示法，置右爲後序表示法。

2. 方法二：建立相對應之二元樹，利用前序追蹤可得前序表示法，利用後序追蹤可得後序表示法。



綜合整理

前序與後序表示法較中序為佳，這是因為前序與後序沒有左、右結合性及優先權之考量，掃瞄一次即可求算結果。而前序與後序表示法中又以後序為最好，因為後序式使用之堆疊（stack）數目為一個，較前序式少。最後，要決定唯一的一顆二元樹（tree），需要中序及後序或中序及前序這兩種組合才行。

END

7.5 二分搜尋樹及其應用

重要性：◈◈◈◈◈

一、二分搜尋樹（Binary Search Tree）

二分搜尋樹是一個二元樹。它可能是空的，若不是空的，則它具備下列特性：

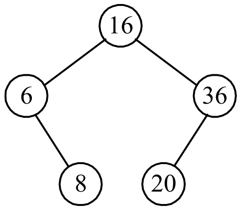
一、二分搜尋樹中每個元素都有一個惟一的鍵值（unique key value），即任兩個元素不會有相同的鍵值。

二、非空左子樹中所有節點的鍵值必定小於樹根的鍵值。

三、非空右子樹中所有節點的鍵值必定大於樹根的鍵值。

四、左子樹與右子樹也都是二分搜尋樹。

如下圖即爲一例：



二、霍夫曼編碼（Huffman Code）

霍夫曼編碼也是一個二元樹的應用。其目的是爲了要縮短訊息的總長度，採用非固定長度的編碼方式，根據各別符號出現之頻率來編碼，頻率較高者編碼長度較短，頻率較低者編碼長度較長。編碼方式如下：

1. 先找出所有符號的頻率。
2. 合併頻率最低的兩個，頻率相加。
3. 重覆步驟2，合併到只剩下一個爲止。
4. 根據合併的關係，對每一次合併的兩項皆分別配置一個位元，一個配置“0”；另一個配置“1”。

舉例說明如下：

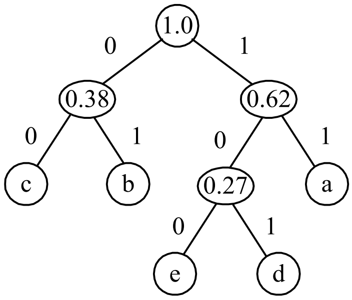
①若一系統有五個符號，其頻率如下表，請編出一套霍夫曼碼。

符號	a	b	c	d	e
頻率	0.35	0.20	0.18	0.15	0.12

②若有一訊息0110110011000011，請以①之編碼方式，將其解碼。

【解】

①



②bdeacca (01 101 100 11 00 00 11)

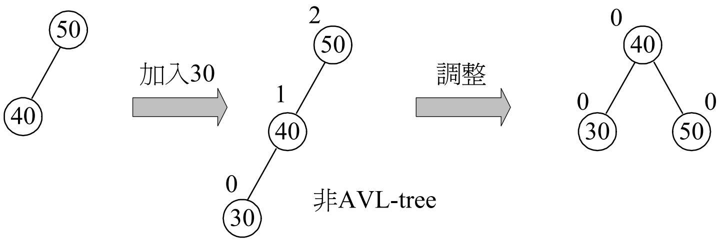
三、平衡二元搜尋樹（AVL Tree）

AVL Tree爲一顆平衡之二元搜尋樹（Balanced Binary Search Tree），若不爲空，則滿足：

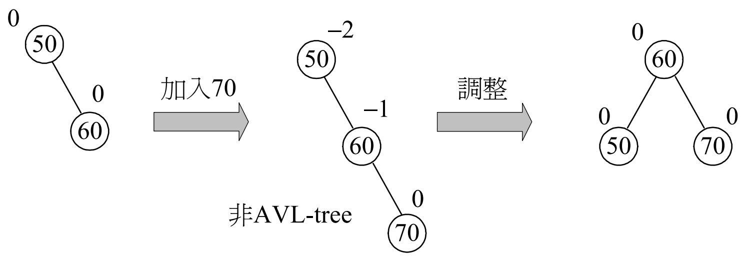
1. 對樹中所有節點而言，其左子樹與右子樹之樹高差皆不大於1，則稱此樹爲一棵平衡樹（balanced tree），寫成數學式子即$|h_L - h_R| \le 1$，h_L，h_R分別爲左右子樹之高度。
2. 滿足左右子樹也都是一顆AVL Tree。

AVL的調整原則爲，先找到**最接近新節點之平衡因子**絕對值大於1（$|BF|>1$）的祖先節點，即平衡因子（balanced factor）不爲+1, 0, −1時，根據旋轉方法的不同，可以將調整情況分成單旋轉和雙旋轉兩類。其中單旋轉有LL和RR兩種：

⑴LL：加入新節點於節點A的左子樹之左子樹

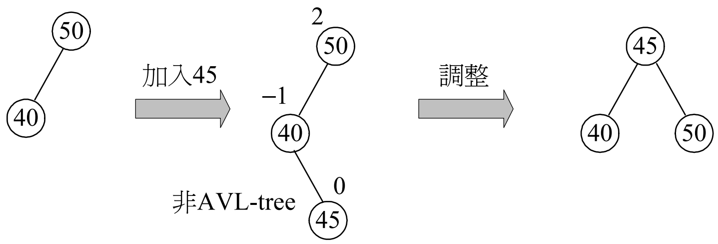


⑵RR：加入新節點於節點A的右子樹之右子樹

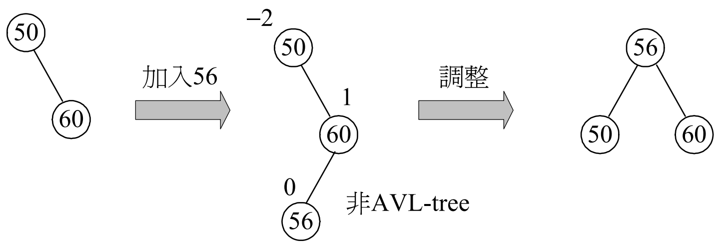


而雙旋轉有LR和RL兩種：

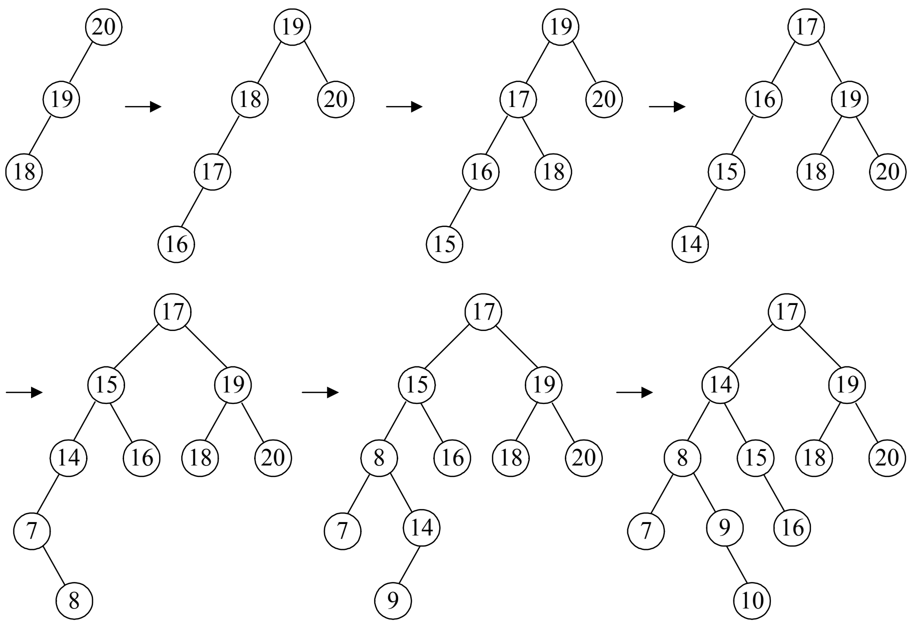
⑴LR：加入新節點於節點A的左子樹之右子樹

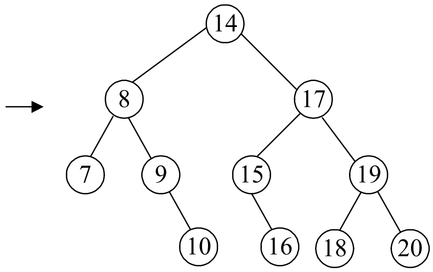


⑵RL：加入新節點於節點A的右子樹之左子樹



⑶



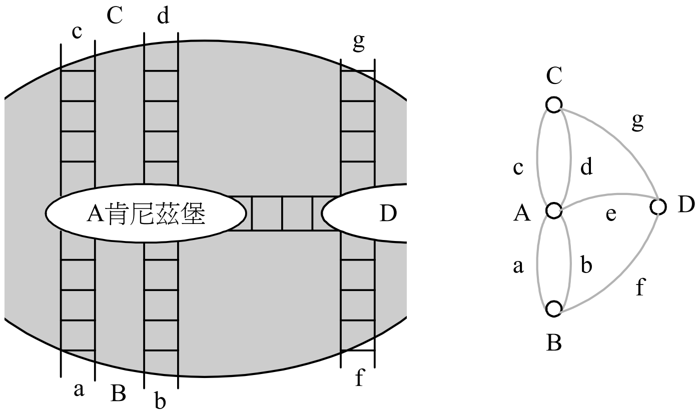


7.6 圖形的資料結構

重要性：◈◈◈◈◈

一、尤拉迴路與尤拉路徑

尤拉迴路與尤拉路徑是圖形理論中，相當有名的名題。它的來源是西元一七三六年，數學家尤拉（Euler）用圖形理論來解決被稱爲「肯尼茲堡橋樑」的問題。在肯尼茲堡城裡面，會有幾個區域間的交通是靠這七座橋，我們的問題是要來決定，若由某個地區出發最後再回到原出發的地區，是否可能把每座橋都走過一次，而且僅能走一次。尤拉的回答是：不可能的。解決的方法是用頂點來表示每個區域，用邊來表示橋樑，對應的圖形如下圖所示：



圖：七橋問題

尤拉得到下列兩個重要的結論：