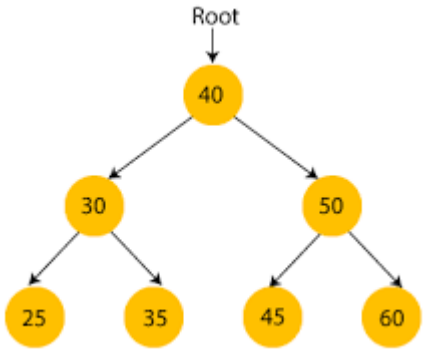


Chapter 07 二元搜尋樹 (Binary Search Tree)

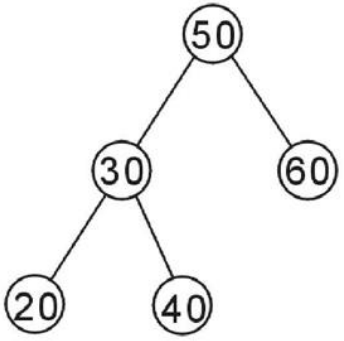
二元搜尋樹是一棵左小右大的二元樹，左子樹的所有鍵值均小於樹根的鍵值。右子樹的所有鍵值均大於樹根的鍵值。

本章綱要

- 二元搜尋樹的定義
- 二元搜尋樹的異動

二元搜尋樹的定義

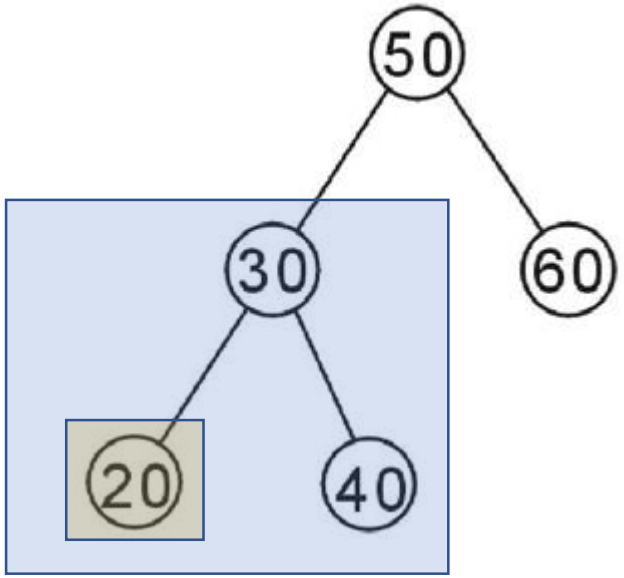
- 二元搜尋樹可以是空集合，假使不是空集合，則樹中的每一節點(node)均含有一鍵值(key value)，而且具有下列特性：
  - 在左子樹的所有鍵值均小於樹根的鍵值。
  - 在右子樹的所有鍵值均大於樹根的鍵值。
  - 左子樹和右子樹亦是二元搜尋樹。
  - 每個鍵值都不一樣。



二元搜尋樹的價值?

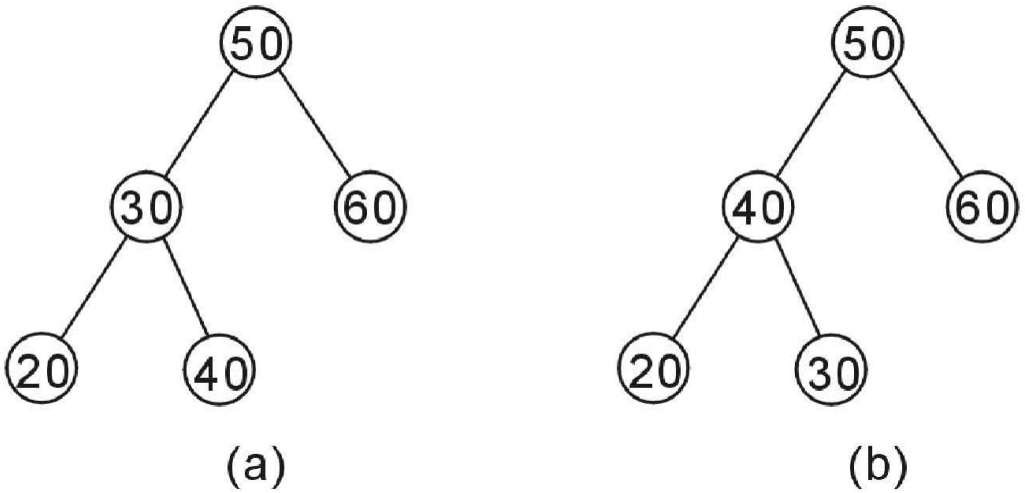
- ✓中序拜訪(左中右)的結果
  - ✓20-30-40-50-60
- ✓從小到大的排序

- 前序拜訪(中左右)的結果
  - 50-30-20-40-60
- 後序拜訪(左右中)的結果
  - 20-40-30-60-50



二元搜尋樹案例

- 下列兩樹是何者為二元搜尋樹



- 其中(a)為一棵二元搜尋樹，而(b)圖形就不是一棵二元搜尋樹，因為鍵值30不應該在鍵值40的右邊。

二元搜尋樹的異動

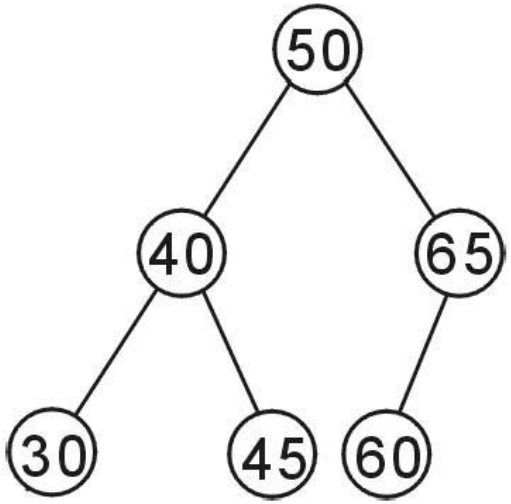
- 二元搜尋樹的異動指的是二元搜尋樹增刪節點，二元搜尋樹增刪節點後，仍然必須維持二元搜尋樹左小右大的特性
- 二元搜尋樹的異動包含
  - 二元搜尋樹的加入節點
  - 二元搜尋樹的刪除節點

二元搜尋樹的加入節點

- 二元搜尋樹的加入和刪除很簡單，因二元搜尋樹的特性是左子樹鍵值均小於樹根的鍵值，而右子樹的鍵值任均大於樹根的鍵值。
- 因此加入某一鍵值只要逐一比較，依據鍵值的大小往右或往左，便可找到此鍵值欲加入的適當位置。

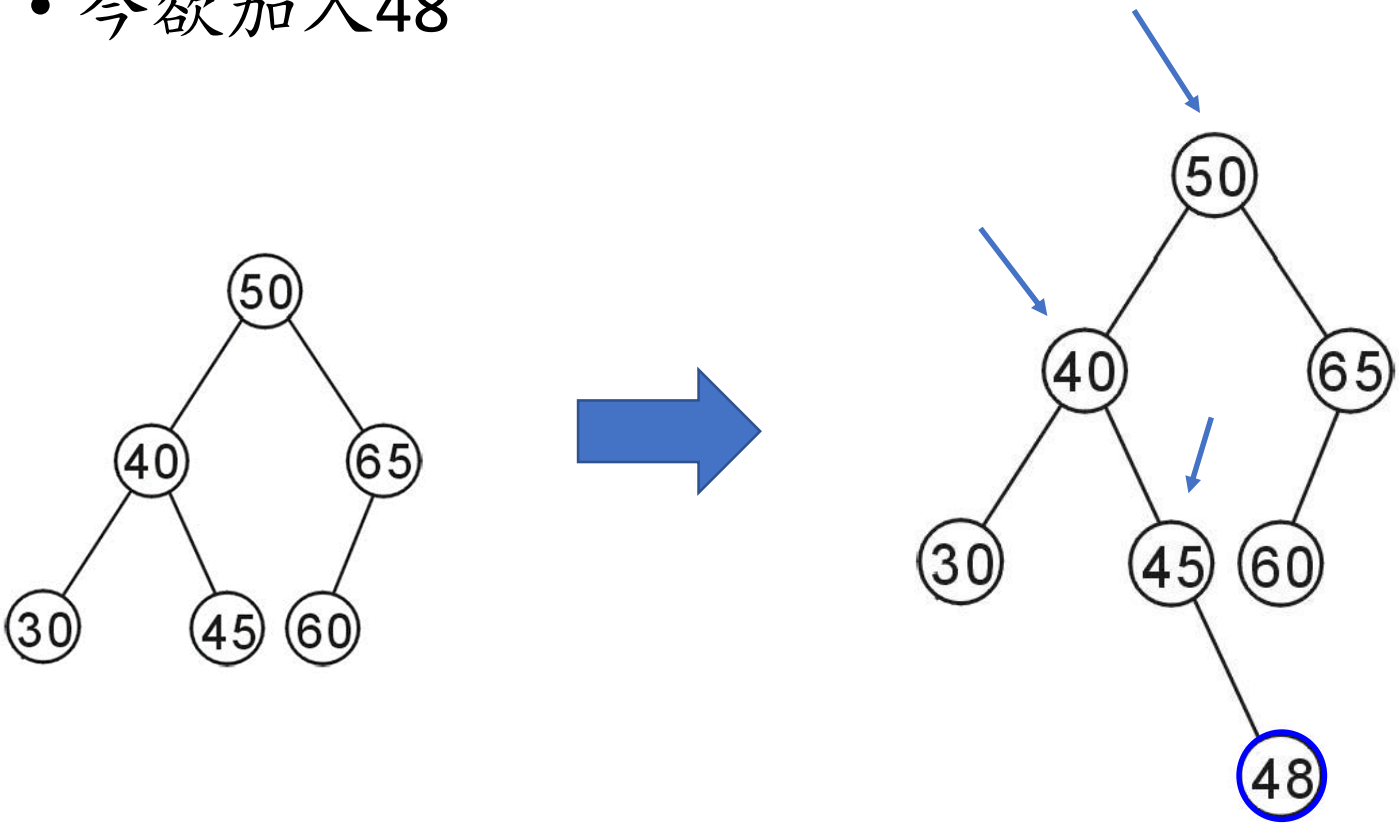
二元搜尋樹的加入節點

- 假設有棵二元搜尋樹如下：



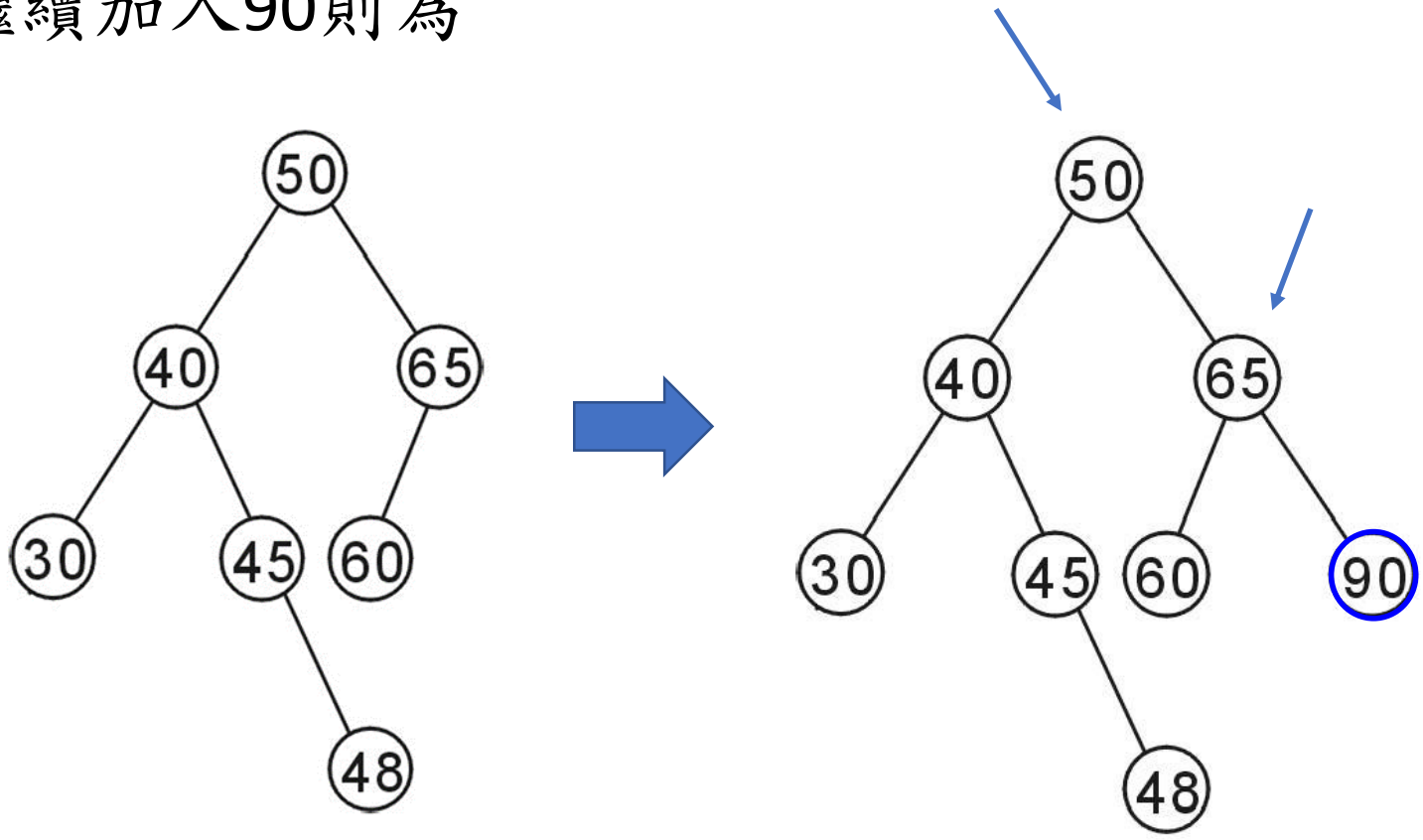
二元搜尋樹的加入節點

- 今欲加入48



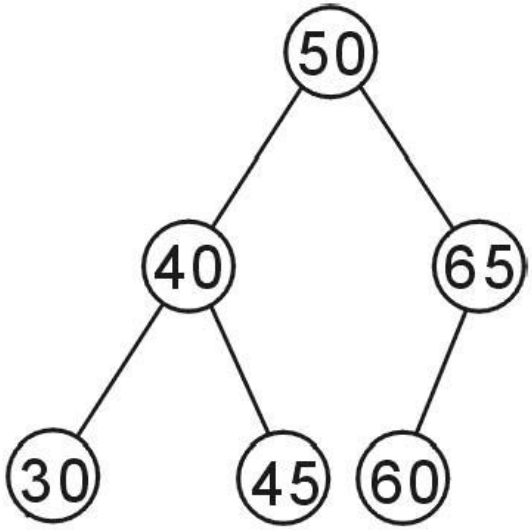
二元搜尋樹的加入節點

- 繼續加入90則為



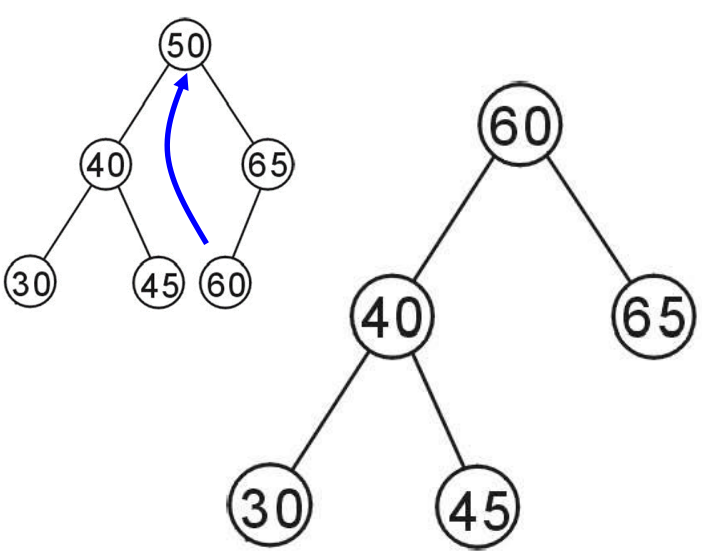
二元搜尋樹的刪除節點

- 刪除某一節點時，若刪除的是樹葉節點，則直接刪除之，假若刪除不是樹葉節點，則在左子樹找一最大的節點或在右子樹找一最小的節點，取代將被刪除的節點，如

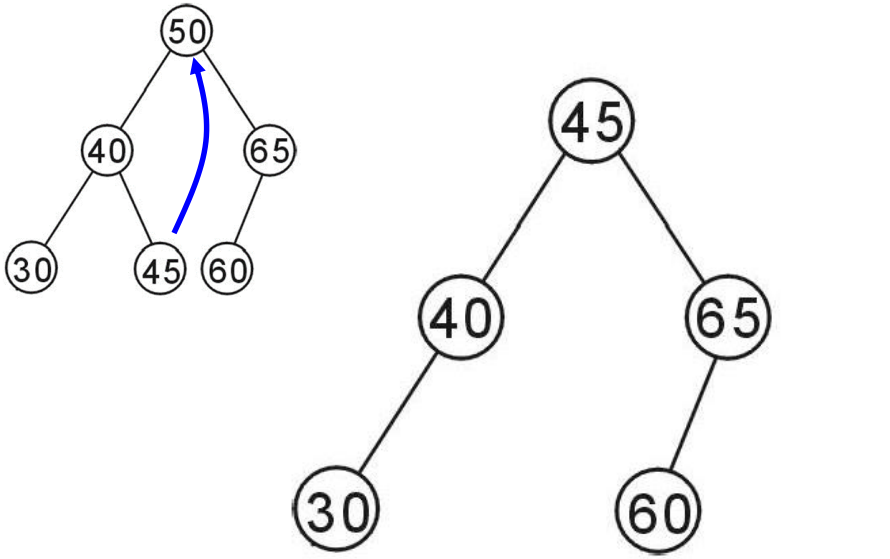


二元搜尋樹的刪除節點

- 刪除50，則可用下列二種方法之一：



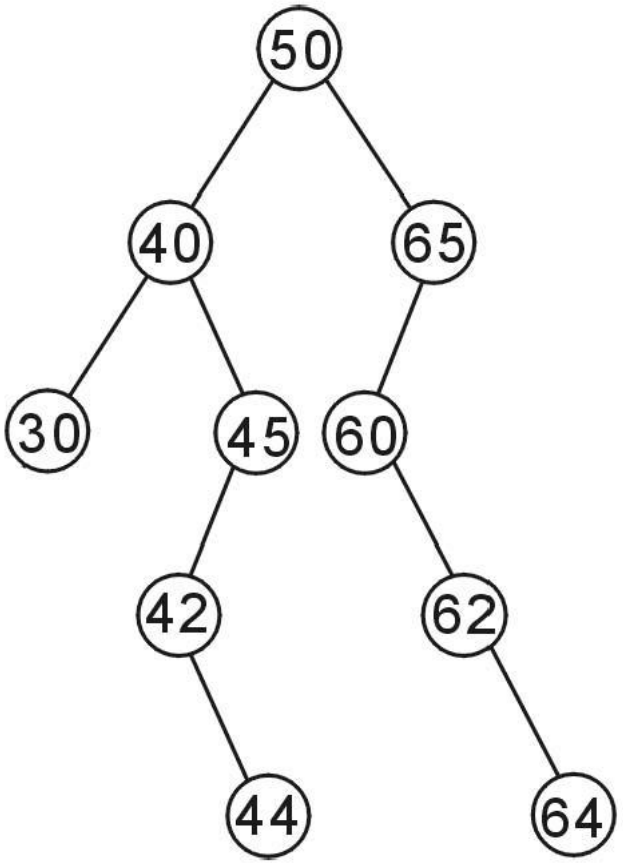
（以右子樹最小的節點取代）



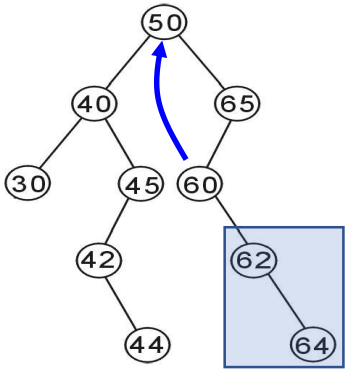
（以左子樹最大的節點取代）

二元搜尋樹的刪除節點

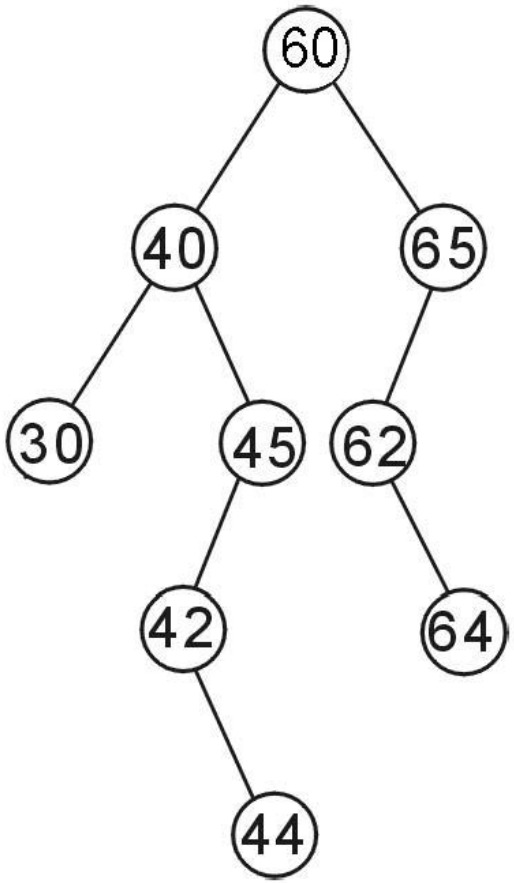
- 若取代的節點有右子樹或左子樹時，則必須加以調整其子節點，如



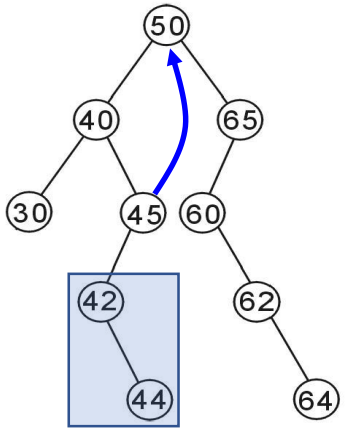
二元搜尋樹的刪除節點



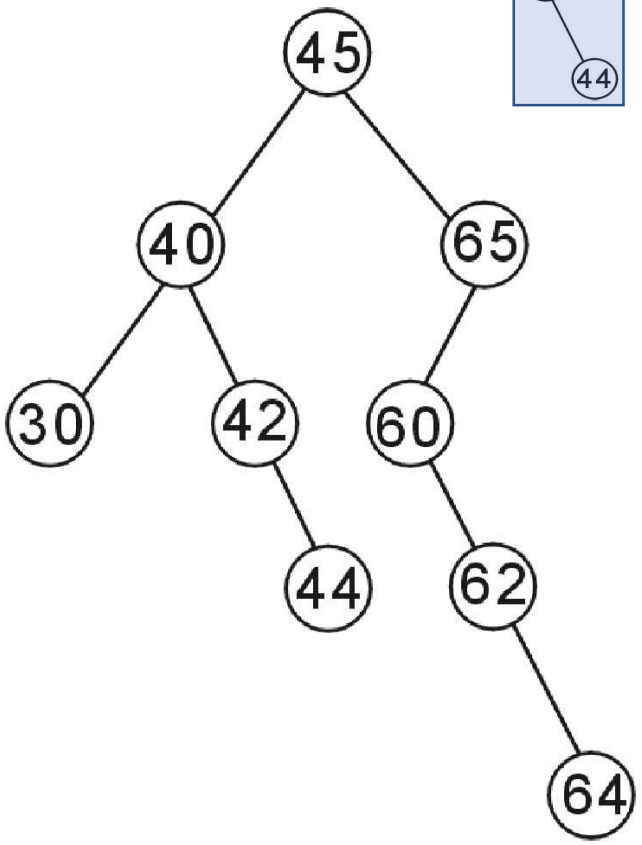
- 今欲刪除50，若以右子樹中最小節點取代刪除節點的話，則60將取代50，但60節點有右子樹（一定不會有左子樹，為什麼？），此時必須將其右子樹重新放在60節點之父節點(65)的左邊鏈結。



二元搜尋樹的刪除節點



- 反之，若以左子樹的最大節點來取代刪除節點的話，則45將取代50，但45此時有左子樹（一定不會有右子樹，為什麼？），此時必須將其左子樹重新放在45節點之父節點(40)的右邊鏈結上。



本章完結