

定義 node 除了基本的 data、lchild、rchild 又多出一個平衡因子 bf，  
左高 bf=1 等高 bf=0 右高 bf=-1

### 插入分 6 種狀況

1. 插入在左子樹 原本左比右高 須做左平衡處理
2. 插入在左子樹 原本左右等高 現在因左子樹增高而樹增高
3. 插入在左子樹 原本右比左高 現在等高
4. 插入到右子樹 原本左比右高 現在等高
5. 插入到右子樹 原本左右子樹等高 現在因右子樹增高而樹增高
6. 插入到右子樹 原本右子樹比左子樹高 須作右平衡處理

leftbalance 中 :T 指向bf值 $\geq |2|$ 的node;L 指向 T 的lchild ;Lr 指向 L 的rchild  
裡面又分成 2 種狀況:

1. 新節點插在 T 的左孩子的左子樹上 作右旋轉，調整 bf 值  $(*T) \rightarrow bf = L \rightarrow bf = 0$
2. 新節點插在 T 的左孩子的右子樹上 作雙旋轉，調整 bf 值：

```
switch(Lr->bf)
{
    case LH:
        (*T)->bf=-1;  L->bf=0;
    case EH:
        (*T)->bf=L->bf=0;
    case RH:
        (*T)->bf=0;L->bf=1;    }
```

右旋轉和左旋轉也分別寫兩個函式 `r_rotate` 和 `l_rotate`，AVLtree 就完成了  
工作站執行結果

[illegible]

