

## 物件導向程式設計 HW6

0416039 李佳燕

1. Increment Operator : inorder 的排序是左子樹->樹根->右子樹，故要找下一個 Node 時，先判斷 CurrentNode->Right 是不是 nullptr，如果不是的話 CurrentNode 先變成 CurrentNode->Right，然後以那個 Node 為樹根，找出左子樹中最下面的 Node，這個 Node 就是我們要找的下一個 Node。如果 CurrentNode->Right 是 nullptr 的話就要往 Parent 找，找出一個 Node 的左子樹中有 CurrentNode，這個 Node 就是答案。如果沒有找到這個 Node 的話，就指定 CurrentNode=nullptr。
2. Decrement Operator : 要找前一個 Node 時，先判斷 CurrentNode->Left 是不是 nullptr，如果不是的話 CurrentNode 先變成 CurrentNode->Left，然後以那個 Node 為樹根，找出右子樹中最下面的 Node，這個 Node 就是我們要找的前一個 Node。如果 CurrentNode->Left 是 nullptr 的話就要往 Parent 找，找出一個 Node 的右子樹中有 CurrentNode，這個 Node 就是答案。如果沒有找到這個 Node 的話，就指定 CurrentNode=nullptr。
3. Find : 如果 `Comp ( KeyToFind , ( CurrentNode -> Mapping ) . first )` 為真的話，表示 KeyToFind 在 CurrentNode 之前，所以就往它的前一個找 `( Find(&(CurrentNode--), CurrentNode, std::forward < key_type > ( KeyToFind ) , Comp) )`，如果不為真，就判斷 `Comp (( CurrentNode -> Mapping ) . first , KeyToFind )` 為不為真，如果為真的話表示 CurrentNode 在 KeyToFind 之前，所以就往它的下一個找 `( Find(&(CurrentNode++), CurrentNode, std::forward<key_type>(KeyToFind), Comp) )`，如果皆不為真的話就表示他們相等(找到了！)，return CurrentNode，如果找到 nullptr 表示 KeyToFind 不在這個 Tree 裡面，return nullptr。
4. Right Rotate : 先把 ThisNode 的左子樹指定給自己設的變數(one)，然後 ThisNode 的左子樹接 one 的右子樹，如果 one 的右子樹不是 nullptr 的話就把 one 右子樹的中 Parent 改為 ThisNode，然後 one 的右子樹就可以接 ThisNode 了，最後把 one 的 Parent 設為 ThisNode 原本的 Parent，ThisNode 的 Parent 設為 one 再更改 ThisNode 和 one 的高度(判斷他們的左子樹比較高還是右子樹，把高度設為比較高的子樹的高度+1) 然後 return one。
5. Left Rotate : 先把 ThisNode 的右子樹指定給自己設的變數(one)，然後 ThisNode 的右子樹接 one 的左子樹，如果 one 的左子樹不是 nullptr 的話就把 one 左子樹的中 Parent 改為 ThisNode，然後

one 的左子樹就可以接 ThisNode 了，最後把 one 的 Parent 設為 ThisNode 原本的 Parent，ThisNode 的 Parent 設為 one 再更改 ThisNode 和 one 的高度(判斷他們的左子樹比較高還是右子樹，把高度設為比較高的子樹的高度+1) 然後 return one。