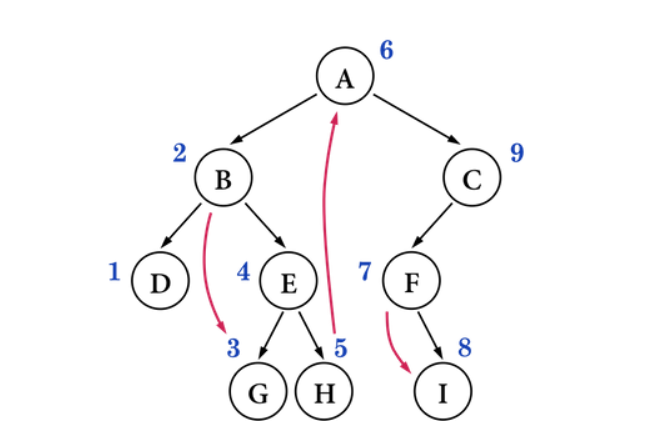
1. BST
2. inorder successor : 對一棵樹執行inorder travsersal後所獲得的序列中，從中任意挑選一節點A，其下一個節點稱之 → 進行BST Delete時用來代替欲刪除的節點



例 : 中序追蹤為 → DBGEHAFIC，A為H的inorder successor

1. find successor code

TreeNode\* InorderSuccessor(TreeNode \*current) {

if (current->leftchild == nullptr && current->rightchild == nullptr) {

while (current == current->parent->rightchild) {

current = current->parent;

}

return current->parent;

} else {

return leftmost(current->rightchild);

}

}

TreeNode\* leftmost(TreeNode \*current) {

if (current == nullptr) {

return nullptr;

} else {

while (current->leftchild != NULL) {

current = current->leftchild;

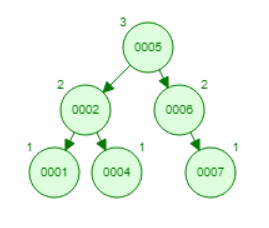
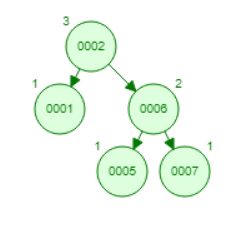
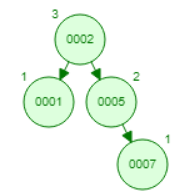
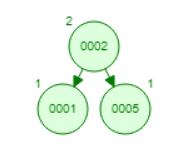
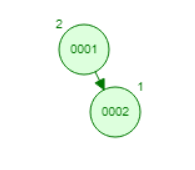
}

return current;

}

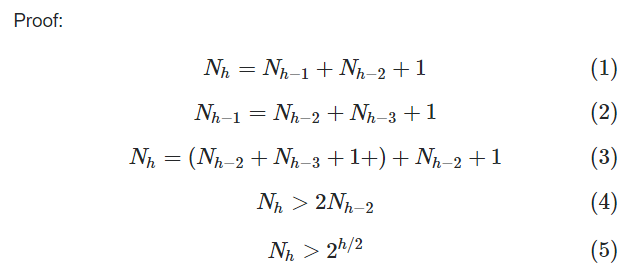
}

1. AVL
2. insert 1 , 2, 5, 7, 6, 4, 3



1. 證明儲存N個key的avl tree高度為o(logn)

→ Nh為形成高度h的avl tree所需的最少節點，因其左右子樹也皆為avl tree高度最多相差1，因此可得以下式子



各取log → 2logN > h，因此其高度為o(logN)