Hw2 bst

Insertion: 將新來的值存進一個 tmp 的 node，若 root == null ，則代表 tree 還沒有東西，所以將 head、tail 和 root 指向 tmp，之後的 insert，根據大小，往左子樹或右子樹走，直到碰到 isthreadl 或 isthreadr，碰到後，將該點的右(左)子樹指向 tmp，然後將 tmp 的左(右)子樹指向該點，然後將 tmp 的右(左)子樹指向該點原本指向的右(左)，並將該點的 isthreadr(isthreadl)設為 0，若 tmp 指向的右(左) == null，則將 tmp 設為 tail(head)。

Deletion:先找到值等於要刪除的點，將 find 指向他，然後找到左(右)子樹的最右(左)，將 idx 指向他，判斷 idx 是不是 leaf，若是，則直接將 idx 的值取代 find 的值，若否，看 find 有幾個子樹，若只有一個，則直接將 find 的 parent 指向 find 的指標改為指向 find 的子樹，並將指向 find 的 thread 改為指向 find 的 thread 指向的點，若有兩個，因為 idx 必只有一子樹，所以可以將 idx 的值取代 find，然後將 idx 的 parent 指向 idx 的指標改為指向 idx 的子樹，再將原本指向的 thread 改為 idx 的 thread 指向的點，此外，若發現刪除的點為 head 或 tail，則將 head 或 tail，改為指向 find

的 parent。

Inorder run:從 head 開始，印出該點的值，並一直向右子樹走，直到碰到 NULL，途中若有碰到 thread，下一次必須向右走但不印值，然後走到最左，重複一開始的動作。

Reverse Inorder run: 從 tail 開始，印出該點的值，並一直向左子樹走，直到碰到 NULL，途中若有碰到 thread，下一次必須向左走但不印值，然後走到最右，重複一開始的動作。