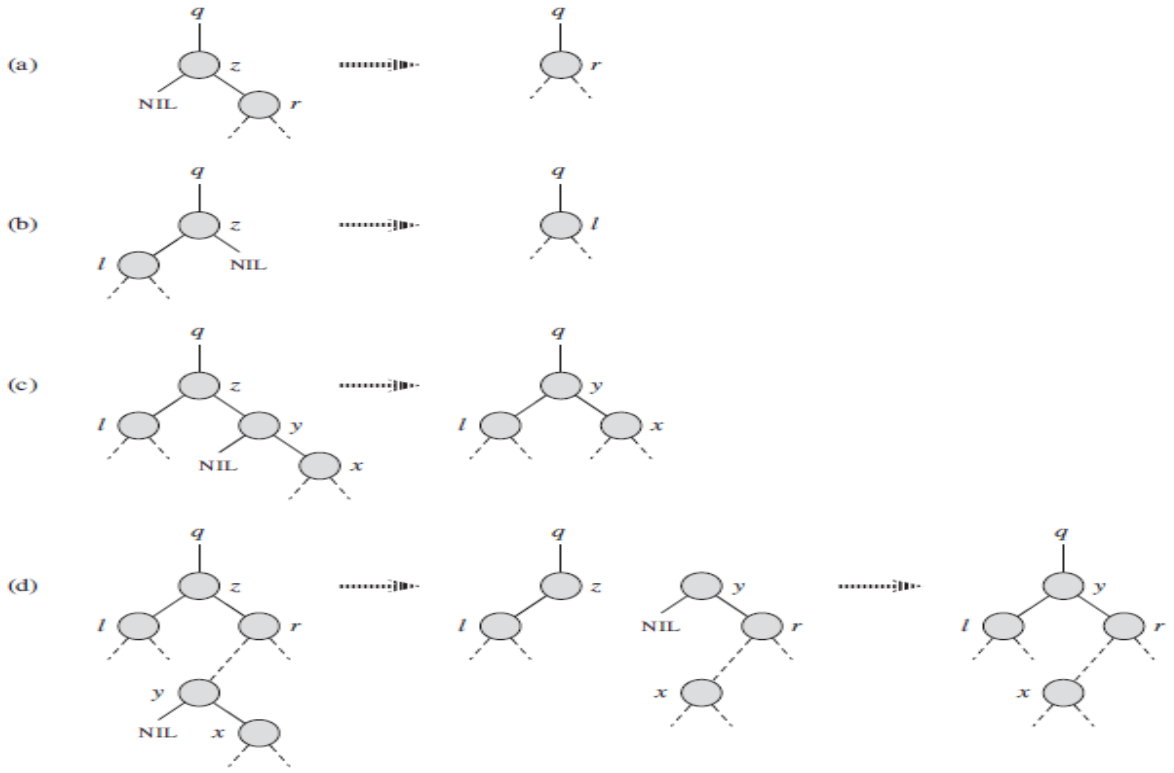
LAB 6 I500/CSCI 609

Binary Search tree Delete

The procedure for deleting a given node z from a binary search tree T takes as arguments pointers to T and z. It organizes its cases a bit differently from the three cases outlined previously by considering the four cases shown in Figure.

- If z has no left child (part (a) of the figure), then we replace z by its right child, which may or may not be NIL. When z's right child is NIL, this case deals with the situation in which z has no children. When z's right child is non-NIL, this case handles the situation in which z has just one child, which is its right child.
- If z has just one child, which is its left child (part (b) of the figure), then we replace z by its left child.
- Otherwise, z has both a left and a right child. We find z's successor y, which lies in z's right subtree and has no left child . We want to splice y out of its current location and have it replace z in the tree.
    a) if y is z's right child (part (c)), then we replace z by y, leaving yzs right child alone.
    b) Otherwise, y lies within zzs right subtree but is not zzs right child (part (d)).

In this case, we first replace y by its own right child, and then we replace z by y.



Picture taken from Cormen etal. Algorithms 3rd edition

```
delete(X, z){
if(z = NULL) //nothing to do
return
if(X < z.data)
delete(X, z.leftChild)
else if(X > z.data)
delete(X, z.rightChild)
else { // found the z to be deleted! Take action based on number of z children
              if(z.left Child = NULL and z.rightChild = NULL){
             delete z
             z = NULL
             return
     }
else if(z.leftChild = NULL){
          tempZ = z
          z = z.rightChild
          delete tempZ
           }
else if(z.rightChild = NULL){
(similar to the case when z.leftChild = NULL)
        }
else {
//replace z.data with minimum data from right subtree
tempnode = findMin(z.rightChild)
z.data = tempnode.data
delete(z.data,z.rightChild)
}
}
```

Problem : Complete the insert, Search ,delete ,print function btree.c file. Also use switch case to insert ,delete and display your data. Print the tree in Inorder after deletion.You can also check the pseudo code given by prof Haixu on oncourse.