8. Let t be an arbitrary binary tree represented using the node structure for a leftist tree.

   a) Write a function to initialize the *shortest* data member of each node in t.

   b) Write a function to convert t into a leftist tree.

   c) What is the complexity of each of these two functions?

(a)
```
int find(node* n) {
    if (n == NULL){
        return 0;
    }
    int l = find(n->lchild), r = find(n->rchild);
    n->shortest_data = min(l, r) + 1;
    return min(l, r) + 1;
}
```
(O(n))

(b)
```
void binary_to_leftist(node* root)
    queue<node*> q
    for n node:
        q.push(node[i]);
    while(!q.empty())                    (n)
        node* n1 = q.pop(), n2 = q.pop();
        n1 = merge_two_leftist_tree(n1, n2);
                                         (log n)
        q.push(n1);
```
(O(n log n))