Exercise sheet 9

Question 9.1

1. Base step: if the tree is empty, then there are 0 internal nodes
if the tree only has the root, then there are $2^0-1=1$ internal nodes
$\Rightarrow$ The proposition holds for $h=0$

Induction: when the proposition holds for $h-1$, then it has $2^{h-1}-1$ internal nodes; if a complete binary tree is of height h, then it has $1+(2^{h-1}-1)\times 2 = 2^h-1$ internal nodes

Conclusion: for $h=0$ it holds, then it holds for any h.

2. Base step: A tree of height 0 is full, with 1 leaf and 0 internal node, it holds.

Induction: Assume there is a full tree that satisfies the proposition. Then we want to add nodes and ensure it's still full. Every operation, we need to find a leaf, and add left child and right child for it.

Originally, there are $n+1$ leaves and n internal nodes.
A leave becomes an internal node, then add 2 leaves.
leaves: $n+1-1+2 = n+2$ internal nodes: $n+1$
Then the proposition still holds.

Conclusion: After every add-nodes operation to a tree with only root node, the proposition still holds.

3. Base step: A tree of height 0, $|V|=1$, $|E|=0$

$\Rightarrow$ satisfy $|V|=|E|+1$

Induction: Assume that we have a tree satisfy $|V|=|E|+1$, for every operation we add a node, which let $|V'|=|V|+1$, $|E'|=|E|+1$, then we still have $|V'|=|E'|+1$

Conclusion: for any tree that is nonempty, $|V|=|E|+1$

Question 9.2

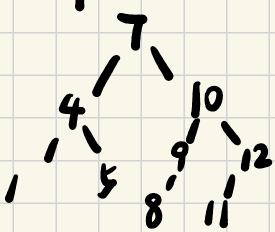
1. start from root 7

11 > 7, move to 7's right child 10

11 > 10, move to 10's right child 12

11 < 12, 12's left child is NIL

11's parent is 12, 12's left child is 11



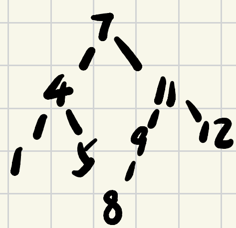
2. 10's right child is 12

12's left child is 11

11 has no left child

replace 11 by NIL

replace 10 by 11



3.

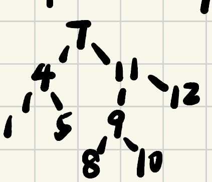
start from root 7

10 > 7, move to 7's right child 11

10 < 11, move to 11's left child 9

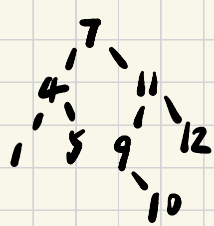
10 > 9, 9's right child is NIL

10's parent is 9, 9's right child is 10



4. 8's left child is NIL

delete 8 directly



5. find y = TREE-MINIMUM(Z.right) = 9

9 is not 7's right child

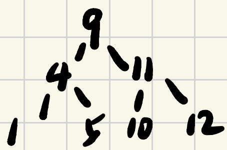
do TRANSPLANT: 9 has parent and 9 has no right child

9's right child is 11, 11's parent is 9

do TRANSPLANT: 7's parent is NIL

then T.root becomes 9

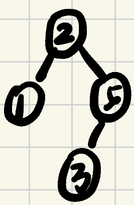
9's left child is 4, 4's parent is 9



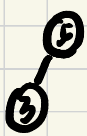
Question 9.3

Yes, they can be different.

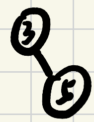
counterexample



delete 1 then 2:



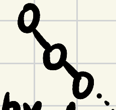
delete 2 then 1



Question 9.4

```
TREE-PREDECESSOR(x)
if x.left ≠ NIL
    return TREE-MAXIMUM(x.left)
else
    y = x.p
    while y ≠ NIL and x == y.left
        x = y
        y = y.p
    return y
```

Question 9.5

Worst case : if all numbers are sorted, then each number has only right child.  After each insertion, the height of tree increases by 1.

runtime : $\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \Theta(n^2)$

inorder tree walk : $\Theta(n)$

$\Rightarrow$ total time : $\Theta(n^2)$

Best case : binary search tree is a balanced tree, the height of the tree is $O(\log n)$, each insertion takes $O(\log n)$ time.

runtime : $O(n\log n)$

inorder tree walk : $\Theta(n)$

$\Rightarrow$ total time : $O(n\log n)$