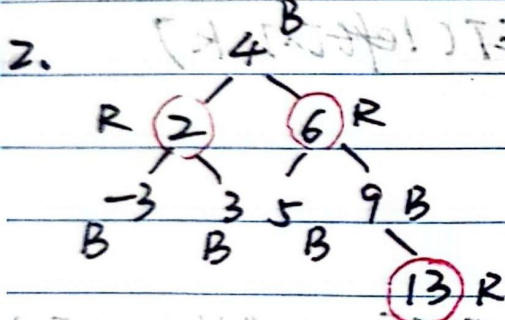
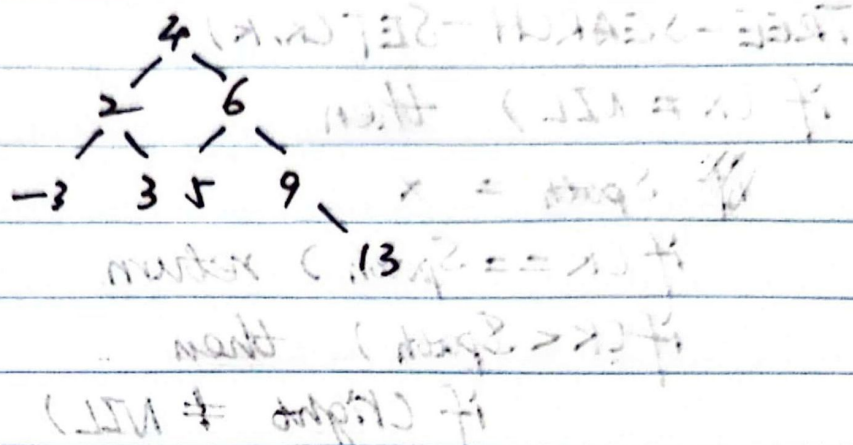
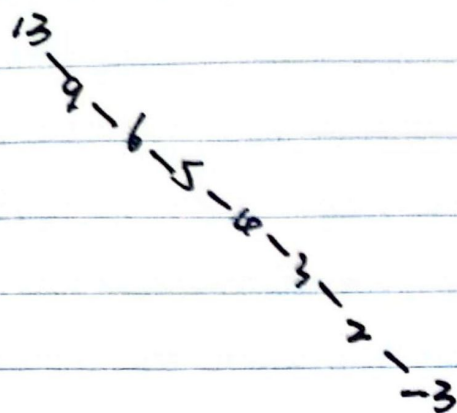
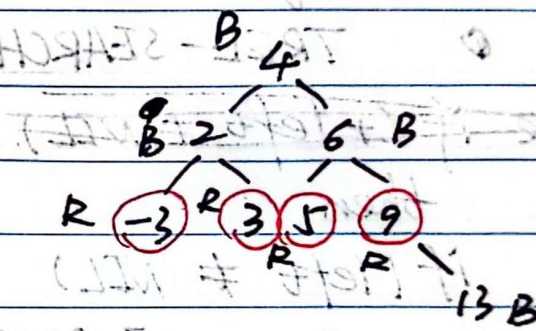


算法hw3

2. max = 8 min = 4



hblack = 2



hblack = 3

Q2 1. if  $x \neq NIL$  then

INORDER-TREE-WALK-LEAF (leaf [x])

if (left[x] == NIL && right[x] == NIL)

print key[x]

INORDER-TREE-WALK-LEAF (right [x])



## 2. ~~Tree Search Set~~ $(x, k)$

$\text{TREE-SEARCH-SET}(x, k)$

if  $(x \neq \text{NIL})$  then

~~if~~  $\text{Spath} = x$

if  $(k == \text{Spath})$  return

if  $(k < \text{Spath})$  then

if  $(\text{right} \neq \text{NIL})$

$\text{Sright} = \text{INORDER-TREE-WALK}(\text{right}[x])$

①  $\text{TREE-SEARCH-SET}(\text{left}[x], k)$

~~else if  $(\text{left} \neq \text{NIL})$~~

else then

if  $(\text{left} \neq \text{NIL})$

$\text{Sleft} = \text{INORDER-TREE-WALK}(\text{left}[x])$

$\text{TREE-SEARCH-SET}(\text{right}[x], k)$

## 3. $(x, k)$ $\text{INORDER-TREE-WALK-LEFT}(x, k)$

if  $(\text{left} \neq \text{NIL})$  &  $\text{left} < k$

print  $\text{key}[x]$

$\text{INORDER-TREE-WALK-LEFT}(\text{right}[x], k)$



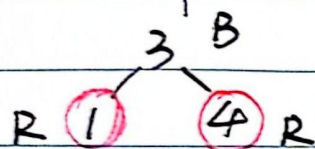


4. (1) (2) T (3) F

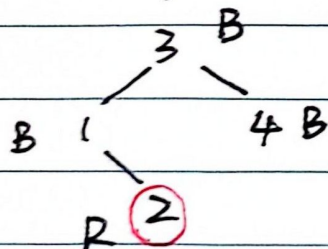
23

1. 有  $n$  个节点的二叉树最多需  $n-1$  次右旋变为一棵每个节点仅有右孩子的树 (从平衡树开始), 则从该树至多  $n-1$  次变为平衡二叉树, 即 ~~至多~~  $n$  个节点的二叉树变为  $n$  个节点的平衡二叉树至多  $2n-2$  次, 即  $O(n)$

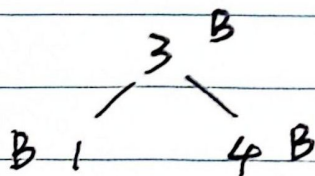
24 不一样



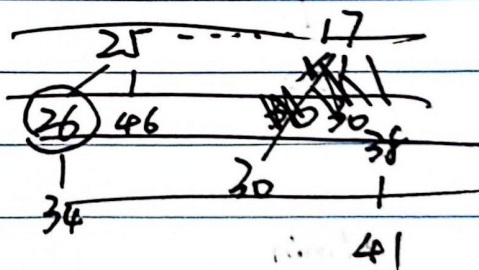
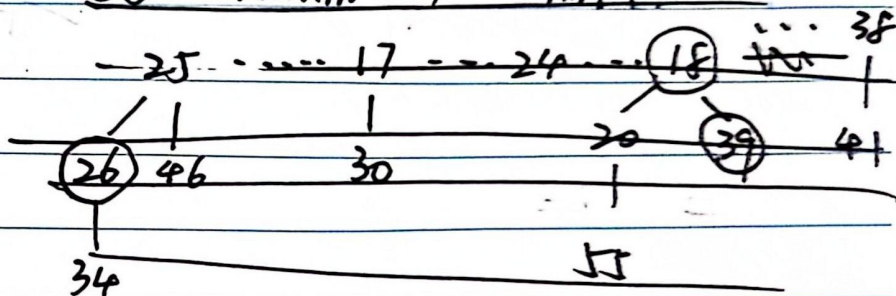
↓ INSERT 2

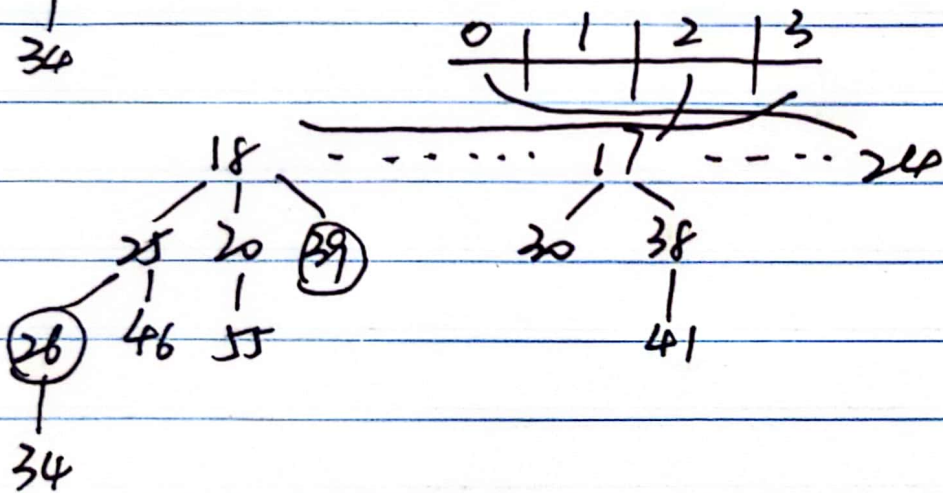
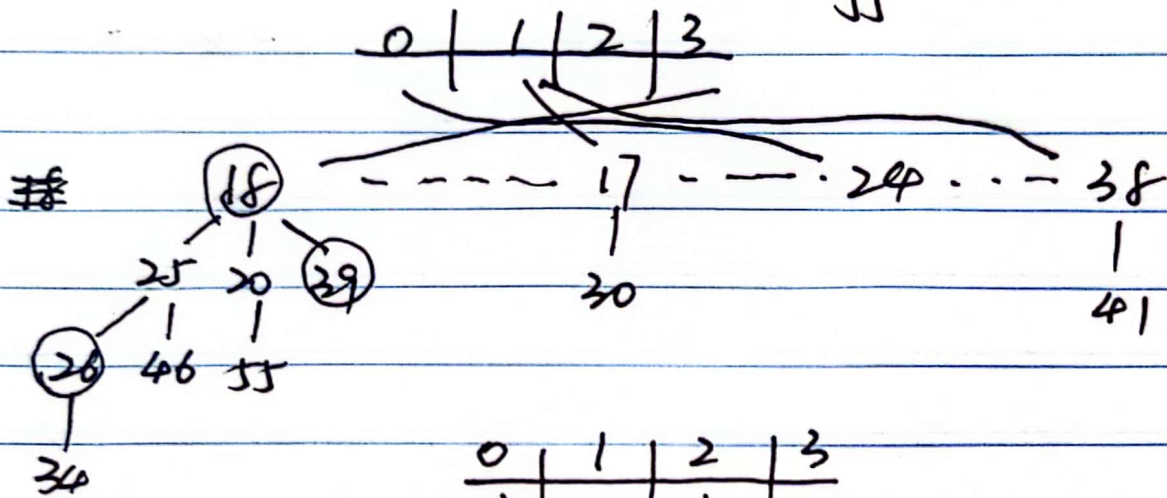
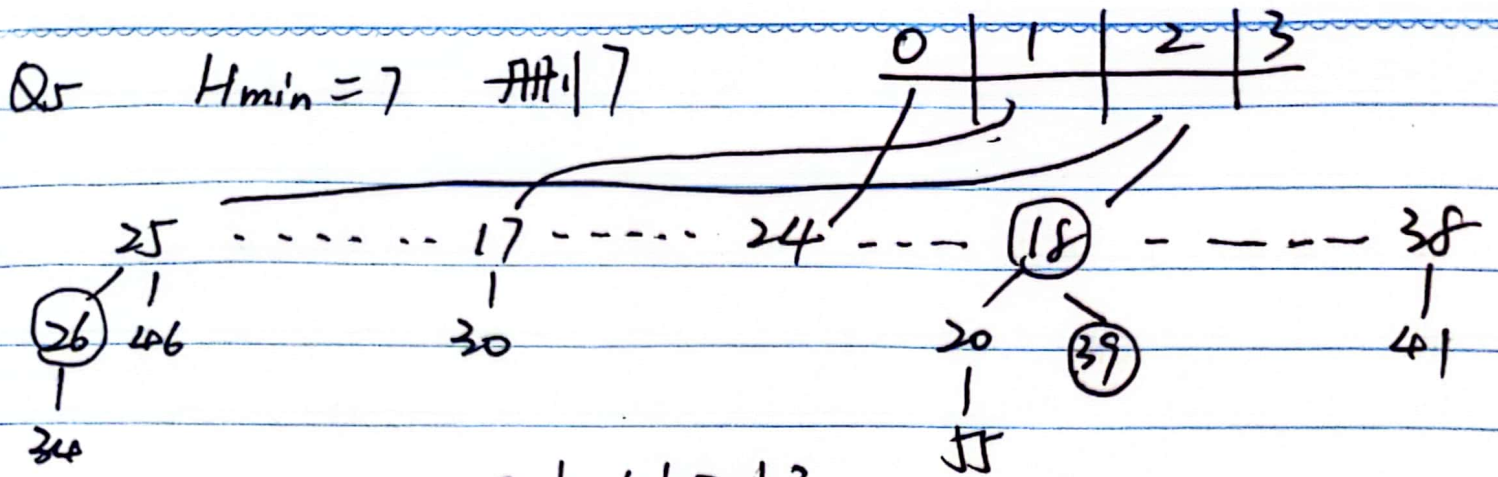


↓ DELETE 2



25. H.min = 7 删 7





$H_{min} = 17$

