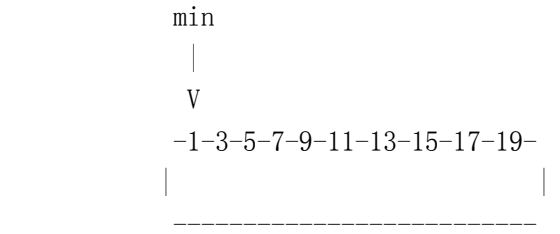
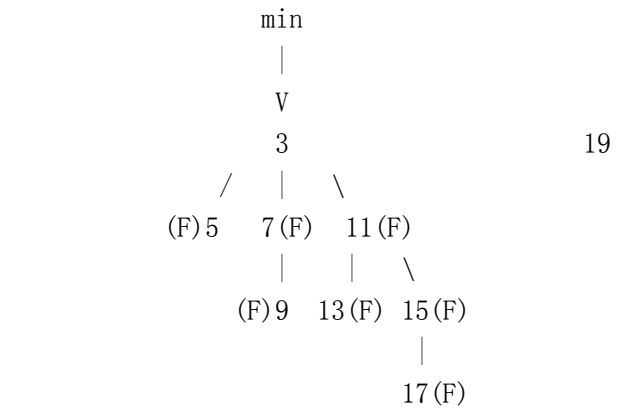
COP 5536/AD 711R Advanced Data Structures
Spring, 2001

Sample Solutions - Exam 2

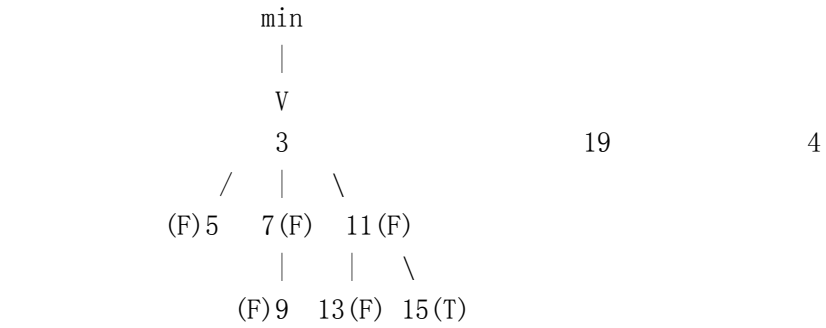
1. (a)




(b) DeleteMin




(c) decrease 17 to 4



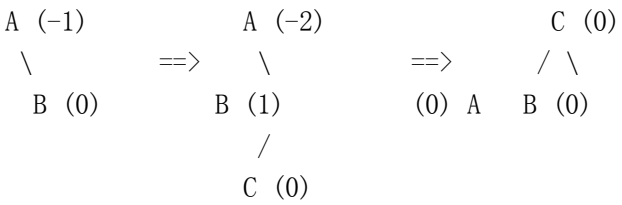

2. treeNode FindKth (ibst, k)
{
if (ibst == null) return null; // not exist
if (ibst->leftSize == k-1)

```
   return (ibst)              // found
else if (ibst->leftSize > k-1) // search left subtree
   return ( FindKth(ibst->leftChild, k))
else
   return ( FindKth(ibst->rightChild, k-(ibst->leftSize+1)));
}
```

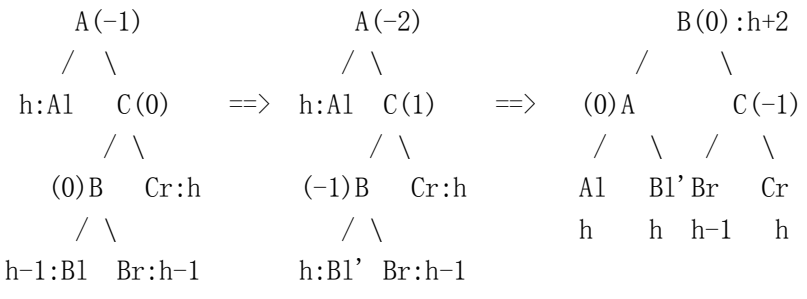
Analysis: Clearly, the algorithm traverses a path from the root to the node containing the k-th smallest key or NULL, error, if k is out of range.
In either case, the length if the path is O(h), where h is the height of the tree.

3. RL rotations
Numbers in parentheses represents balance factors.

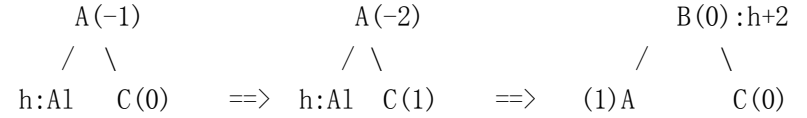
(1) case 1:

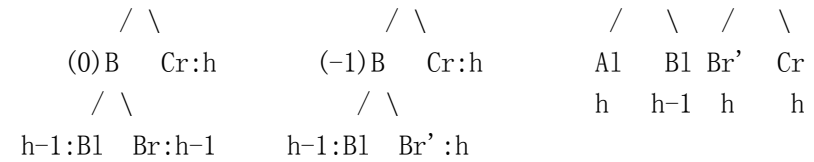



(2) case 2:




(3) case 3:

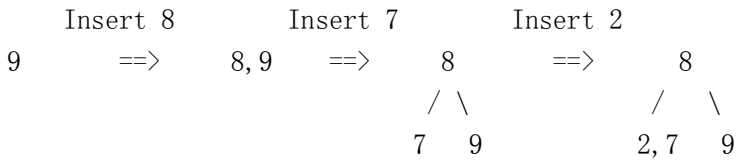


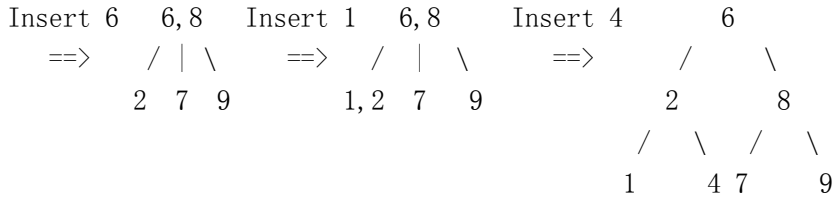


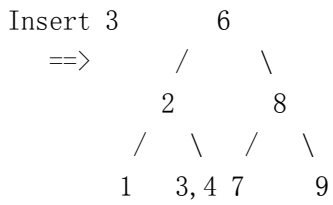

Before Insert to Br After

4.

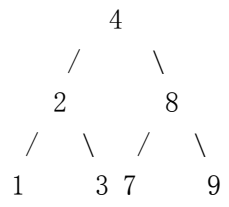
(a)



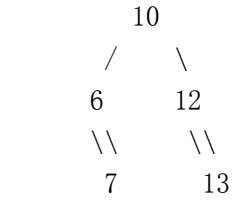

(b) Delete 6



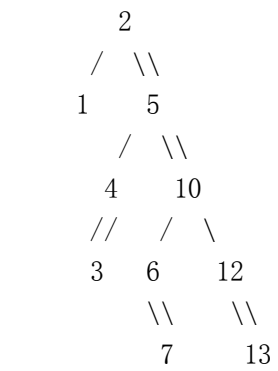

5. (a) Numbers in parentheses represent ranks.

i: Follow the right child pointer until rank(B) = rank(x), where rank(B) =1, x is a node pointer of tree S.

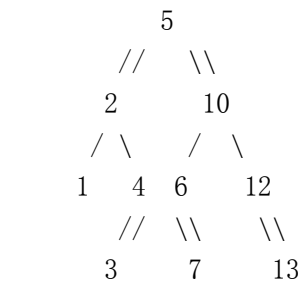
ii: Combine subtree x, 10, and B




iii: Combine the result of ii to node 5 through a red pointer




iv: Perform a RR rotation




(b)

i: Search node 3

ii: Perform Join(B', 4, null)

B': 4

iii: Join (B', 5, right-subtree of node 5)

B': 5

4
6

iv: Join (left-subtree of node 2, 2, S')

S': 

Result: