### 附：例子

public static int sortedSearch (int[] a, int x)

/\*\*@ REQUIRES: (\all int i, j; 0 <= i < j < a.length;  a[i] <= a[j]);

@ MODIFIES: None;

@ EFFECTS:

(\exist int k; a[k] == x) ==> \result == k ;

(\all int k; a[k] != x) ==> \result == -1;

\*/

{

found = false;

for(int i=a.length-1;i>=0;i--){

if(a[i] == x){

z = i;

found = true;

}

if(found) return z; else return -1;

}

}

public static int search (int [ ] a, int x)

/\*\*@ EFFECTS:

(\exist int k; a[k] == x) ==> \result == k ;

(\all int k; 0<=k<a.length; a[k] != x) ==> \result == -1;

\*/

public static void sort (int [ ] a)

/\*\*@ MODIFIES: a;

@ EFFECTS:

(\all int i, j; 0 <= i < j < a.length;  a[i] <= a[j]);

\*/

public void insert(BinarySortedTree tree, int x)

/\*\*@ REQUIRES: (\all BinarySortedTree node in tree; node.left.value <= node.value <=node.right.value);

@ MODIFIES: tree;

@ EFFECTS:

(\old(tree) == null) ==> (tree.root == x);

(\exist BinarySortedTree x\_node in tree, (x\_node.value == x) &&

( (x\_node.left.value != null) ==>( x\_node.left.value <= x\_node.value)) &&

( (x\_node.right.value != null) ==>( x\_node.value <=x\_node.right.value) );

\*/

public int remove(BinarySortedTree tree, int x)

/\*\*@ REQUIRES: (\all BinarySortedTree node in tree; node.left.value <= node.value <=node.right.value);

@ MODIFIES: tree;

@ EFFECTS:

(\all BinarySortedTree node in tree; node.left.value <= node.value <=node.right.value);

(\exist BinarySortedTree x\_node in \old(tree); x\_node.value == x) ==> (\all BinarySortedTree node in tree; node.value != x) && \result == 1;

(\all BinarySortedTree node in \old(tree); node.value != x) ==> \result == -1;

\*/

public static void removeDupls(Vector v)

/\*\*@ REQUIRES: v != null ;

@ MODIFIES: v;

@ EFFECTS:

(\all int i,j ; 0<=i<j<v.length; v.get(i) != v.get(j) );

(\all int i; 0<=i<\old(v).length; v.contains(\old(v).get(i))) ;

\*/

public static void addMax(Vector v, Integer x) throw NullPointerException, NotSmallException

/\*\*@ REQUIRES: (\all int i; 0<=i<v.length; v.get(i) *instanceof* Integer);

@ MODIFIES: v;

@ EFFECTS: *normal\_behavior*

(\all int i ; 0<=i<\old(v).length; \old(v).get(i).intValue <= x.intValue) ==> (v.size == \old(v).size+1) && (v.contains(x)) ;

(\old(v) == null) ==> *exceptional\_behavior* (NullPointerException);

(\exist int i ; 0<=i<\old(v).length; \old(v).get(i).intValue > x.intValue) ==> *exceptional\_behavior* (NotSmallException);

\*/

public static int fact(int n) throws NonPositiveException

/\*\*@ EFFECTS: *normal\_behavior*

\result == (n!);

(n < 0) ==> *exceptional\_behavior* (NonPositiveException);

\*/

public static int min (int[ ] a) throws NullPointerException, EmptyException

/\*\*@ EFFECTS: *normal\_behavior*

\result == \min a;

(a == null) ==> *exceptional\_behavior* (NullPointerException);

(a.length == 0) ==> *exceptional\_behavior* (EmptyException);

\*/