### 附：例子

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

/\*@ REQUIRES: (\all int i, j; 0 <= i & i < j & 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; a[k] != x) ==> \result == -1;

\*/

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

/\*@ REQUIRES: (\all int i, j; 0 <= i & i < j & 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;

\*/

public static void sort (int [ ] a)

/\*@ MODIFIES: a;

@ EFFECTS:

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

e.g., If \old(a) = [3, 1, 6, 1], then a= [1, 1, 3, 6];

\*/

public void insert(BinarySortedTree tree, int x)

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

@ MODIFIES: tree;

@ EFFECTS:

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

(\exist x\_node, (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 tree.node; node.left.value <= node.value <=node.right.value);

@ MODIFIES: tree;

@ EFFECTS:

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

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

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

\*/

public static void removeDupls(Vector v)

/\*@ REQUIRES: v != null ;

@ MODIFIES: v;

@ EFFECTS:

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

(\exist int i,j ; \old(v).get(i) == \old(v).get(j)) ==> (v.size == \old(v).size-1) ;

\*/

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

/\*@ REQUIRES: (\all element in v; element *instanceof* Integer);

@ MODIFIES: v;

@ EFFECTS: *normal\_behavior*

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

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

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

\*/

public static int fact(int n) throws NonPositiveException

//@ EFFECTS: *normal\_behavior*

\result == (n!);

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

{

if(n<=0) *throw* new NonPositiveException(“n in Num.fact”);

... }

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);

{

int m;

try { m = a[0]; } catch (IndexOutOfBoundsException e) {

throw new EmptyException (”Arrays.min"); } for (int i=1; i < a.length; i++)

if (a[i] < m) m=a[i]; return m;

}