

```

39     int find(T x, Node<T>[] preds, Node<T>[] succs) {
40         int key = x.hashCode();
41         int lFound = -1;
42         Node<T> pred = head;
43         for (int level = MAX_LEVEL; level >= 0; level--) {
44             volatile Node<T> curr = pred.next[level];
45             while (key > curr.key) {
46                 pred = curr; curr = pred.next[level];
47             }
48             if (lFound == -1 && key == curr.key) {
49                 lFound = level;
50             }
51             preds[level] = pred;
52             succs[level] = curr;
53         }
54         return lFound;
55     }

```

FIGURE 14.5 The LazySkipList class: the wait-free find() method. This algorithm is the same as in the sequential SkipList implementation. The preds[] and succs[] arrays are filled from the maximum level to level 0 with the predecessor and successor references for the given key.