```
37    boolean add(T x) {
38      int topLevel = randomLevel();
39      int bottomLevel = 0;
40      Node<T>[] preds = (Node<T>[]) new Node[MAX_LEVEL + 1];
41      Node<T>[] succs = (Node<T>[]) new Node[MAX_LEVEL + 1];
42      while (true) {
43        boolean found = find(x, preds, succs);
44        if (found) {
45          return false;
46        } else {
47          Node<T> newNode = new Node(x, topLevel);
48          for (int level = bottomLevel; level <= topLevel; level++) {
49            Node<T> succ = succs[level];
50            newNode.next[level].set(succ, false);
51          }
52          Node<T> pred = preds[bottomLevel];
53          Node<T> succ = succs[bottomLevel];
54          if (!pred.next[bottomLevel].compareAndSet(succ, newNode,
55                                                    false, false)) {
56            continue;
57          }
58          for (int level = bottomLevel+1; level <= topLevel; level++) {
59            while (true) {
60              pred = preds[level];
61              succ = succs[level];
62              if (pred.next[level].compareAndSet(succ, newNode, false, false))
63                break;
64              find(x, preds, succs);
65            }
66          }
67          return true;
68        }
69      }
70    }
```

FIGURE 14.11 The LockFreeSkipList class: the add() method.