```
1  public boolean add(T item) {
2    int key = item.hashCode();
3    while (true) {
4      Node pred = head;
5      Node curr = pred.next;
6      while (curr.key < key) {
7        pred = curr; curr = curr.next;
8      }
9      pred.lock();
10     try {
11       curr.lock();
12       try {
13         if (validate(pred, curr)) {
14           if (curr.key == key) {
15             return false;
16           } else {
17             Node node = new Node(item);
18             node.next = curr;
19             pred.next = node;
20             return true;
21           }
22         }
23       } finally {
24         curr.unlock();
25       }
26     } finally {
27       pred.unlock();
28     }
29   }
30 }
```

FIGURE 9.11 The OptimisticList class: the add() method traverses the list ignoring locks, acquires locks, and validates before adding the new node.