```
16    public void acquire(T x) {
17      boolean[] mark = {true};
18      Thread me = Thread.currentThread();
19      Thread who;
20      while (true) {
21        do { // wait until not resizing
22          who = owner.get(mark);
23        } while (mark[0] && who != me);
24        ReentrantLock[][] oldLocks = locks;
25        ReentrantLock oldLock0 = oldLocks[0][hash0(x) % oldLocks[0].length];
26        ReentrantLock oldLock1 = oldLocks[1][hash1(x) % oldLocks[1].length];
27        oldLock0.lock();
28        oldLock1.lock();
29        who = owner.get(mark);
30        if ((!mark[0] || who == me) && locks == oldLocks) {
31          return;
32        } else {
33          oldLock0.unlock();
34          oldLock1.unlock();
35        }
36      }
37    }
38    public void release(T x) {
39      locks[0][hash0(x)].unlock();
40      locks[1][hash1(x)].unlock();
41    }
```

FIGURE 13.32 RefinableCuckooHashSet<T>: acquire() and release() methods.