

```
1 template <class T>
2 class LockFreeQueue<T> {
3     std::atomic<Node*> head;
4     std::atomic<Node*> tail;
5     ...
6     void enq(T item) {
7         Node* node = new Node(item);
8         while (true) {
9             Node* last = tail;
10            Node* next = last->next;
11            if (last == tail) {
12                if (next == null) {
13                    if (last->next.compare_exchange_strong(next, node)) {
14                        tail.compare_exchange_strong(last, node);
15                        return;
16                    }
17                } else {
18                    tail.compare_exchange_strong(last, next);
19                }
20            }
21        }
22    }
23 }
```

FIGURE 20.2 The LockFreeQueue class: the enq() method.