

```

1  void enq(T item) {
2      Node* node = new Node(item);
3      while (true) {
4          Node* last = tail;
5          Node* next = last->next;
6          if (last == tail) {
7              std::atomic<Node*> target[2] = {&last->next, &tail};
8              Node* expect[2] = {next, last};
9              Node* update[2] = {node, node};
10             if (multiCompareAndSet(target, expect, update)) return;
11         }
12     }
13 }

```

FIGURE 20.4 The LockFreeQueue class: simplified enq() method with multiCompareAndSet().