

```

1  class Node<T>{
2      T item;
3      int key;
4      std::atomic<Node<T>*> next;
5  public:
6      Node() : item(), next(nullptr) { }
7      Node(T _item, int _key, Node* n) : item(_item), key(_key), next(n) { }
8  };
9  // return whether the low bit of the pointer is marked or not
10 bool is_marked(Node* ptr) {
11     return ((uintptr_t)ptr)&1;
12 }
13 // clear the mark bit of a pointer
14 Node* unmark(Node* ptr) {
15     return (Node*)((uintptr_t)ptr & ~(uintptr_t)1);
16 }
17 // set the mark bit of a pointer
18 Node* mark(Node* ptr) {
19     return (Node*)((uintptr_t)ptr | 1);
20 }

```

FIGURE 19.3 C++ nonblocking linked list: node data type and helper functions.