```
1  public class SimpleTree<T> implements PQueue<T> {
2    int range;
3    List<TreeNode> leaves;
4    TreeNode root;
5    public SimpleTree(int logRange) {
6      range = (1 << logRange);
7      leaves = new ArrayList<TreeNode>(range);
8      root = buildTree(logRange, 0);
9    }
10   public void add(T item, int score) {
11     TreeNode node = leaves.get(score);
12     node.bin.put(item);
13     while(node != root) {
14       TreeNode parent = node.parent;
15       if (node == parent.left) {
16         parent.counter.getAndIncrement();
17       }
18       node = parent;
19     }
20   }
21   public T removeMin() {
22     TreeNode node = root;
23     while(!node.isLeaf()) {
24       if (node.counter.boundedGetAndDecrement() > 0 ) {
25         node = node.left;
26       } else {
27         node = node.right;
28       }
29     }
30     return node.bin.get();
31   }
32 }
```

FIGURE 15.3 The SimpleTree bounded-range priority queue.