```
1   public class WorkStealingThread {
2     DEQue[] queue;
3     public WorkStealingThread(DEQue[] queue) {
4       this.queue = queue;
5     }
6     public void run() {
7       int me = ThreadID.get();
8       RecursiveAction task = queue[me].popBottom();
9       while (true) {
10        while (task != null) {
11          task.compute();
12          task = queue[me].popBottom();
13        }
14        while (task == null) {
15          int victim = ThreadLocalRandom.current().nextInt(queue.length);
16          if (!queue[victim].isEmpty()) {
17            task = queue[victim].popTop();
18          }
19        }
20      }
21    }
22  }
```

FIGURE 18.10 Work stealing executor pool revisited.