```
1  class CircularArray {
2      private int logCapacity;
3      private RecursiveAction[] currentTasks;
4      CircularArray(int logCapacity) {
5        this.logCapacity = logCapacity;
6        currentTasks = new RecursiveAction[1 << logCapacity];
7      }
8      int capacity() {
9        return 1 << logCapacity;
10     }
11     RecursiveAction get(int i) {
12       return currentTasks[i % capacity()];
13     }
14     void put(int i, RecursiveAction task) {
15       currentTasks[i % capacity()] = task;
16     }
17     CircularArray resize(int bottom, int top) {
18       CircularArray newTasks =
19           new CircularArray(logCapacity+1);
20       for (int i = top; i < bottom; i++) {
21         newTasks.put(i, get(i));
22       }
23       return newTasks;
24     }
25   }
```

FIGURE 16.13 The UnboundedDEQue class: the circular task array.