```
public RecursiveAction popTop() {
  int oldTop = top.get();
  int newTop = oldTop + 1;
  int oldBottom = bottom;
  CircularArray currentTasks = tasks;
  int size = oldBottom - oldTop;
  if (size <= 0) return null;
  RecursiveAction r = tasks.get(oldTop);
  if (top.compareAndSet(oldTop, newTop))
    return r;
  return null;
}

public RecursiveAction popBottom() {
  int newBottom = --bottom;
  int oldTop = top.get();
  int newTop = oldTop + 1;
  int size = newBottom - oldTop;
  if (size < 0) {
    bottom = oldTop;
    return null;
  }
  RecursiveAction r = tasks.get(newBottom);
  if (size > 0)
    return r;
  if (!top.compareAndSet(oldTop, newTop))
    r = null;
  bottom = newTop;
  return r;
}
```

FIGURE 16.15 The UnboundedDEQue class: popTop() and popBottom() methods.