```
public RecursiveAction popTop() {
  int[] stamp = new int[1];
  int oldTop = top.get(stamp);
  int newTop = oldTop + 1;
  int oldStamp = stamp[0];
  if (bottom <= oldTop)
    return null;
  RecursiveAction r = tasks[oldTop];
  if (top.compareAndSet(oldTop, newTop, oldStamp, oldStamp))
    return r;
  else
    return null;
}
public RecursiveAction popBottom() {
  if (bottom == 0)
    return null;
  int newBottom = --bottom;
  RecursiveAction r = tasks[newBottom];
  int[] stamp = new int[1];
  int oldTop = top.get(stamp);
  int newTop = 0;
  int oldStamp = stamp[0];
  int newStamp = oldStamp + 1;
  if (newBottom > oldTop)
    return r;
  if (newBottom == oldTop) {
    bottom = 0;
    if (top.compareAndSet(oldTop, newTop, oldStamp, newStamp))
      return r;
    }
  top.set(newTop, newStamp);
  return null;
}
```

FIGURE 16.11 The BoundedDEQue class: popTop() and popBottom() methods.