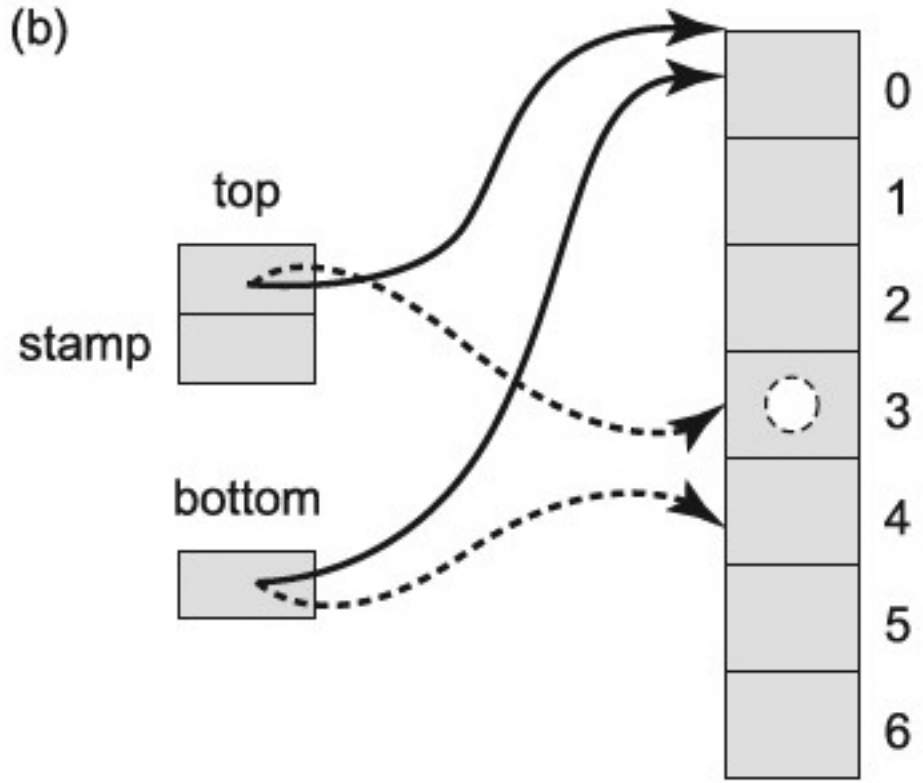

FIGURE 16.12B The BoundedDEQue implementation. In part (a), popTop() and popBottom() are called concurrently while there is more than one task in the BoundedDEQue. The popTop() method reads the element in entry 2 and calls compareAndSet() to redirect the top reference to entry 3. The popBottom() method redirects the bottom reference from 5 to 4 using a simple store and then, after checking that bottom is greater than top, it removes the task in entry 4. In part (b), there is only a single task. When popBottom() detects that, after redirecting from 4 to 3, top and bottom are equal, it attempts to redirect top with a compareAndSet(). Before doing so, it redirects bottom to 0 because this last task will be removed by one of the two popping methods. If popTop() detects that top and bottom are equal, it gives up; otherwise, it tries to advance top using compareAndSet(). If both methods apply compareAndSet() to the top, one wins and removes the task. In any case, win or lose, popBottom() resets top to 0 since the BoundedDEQue is now empty.