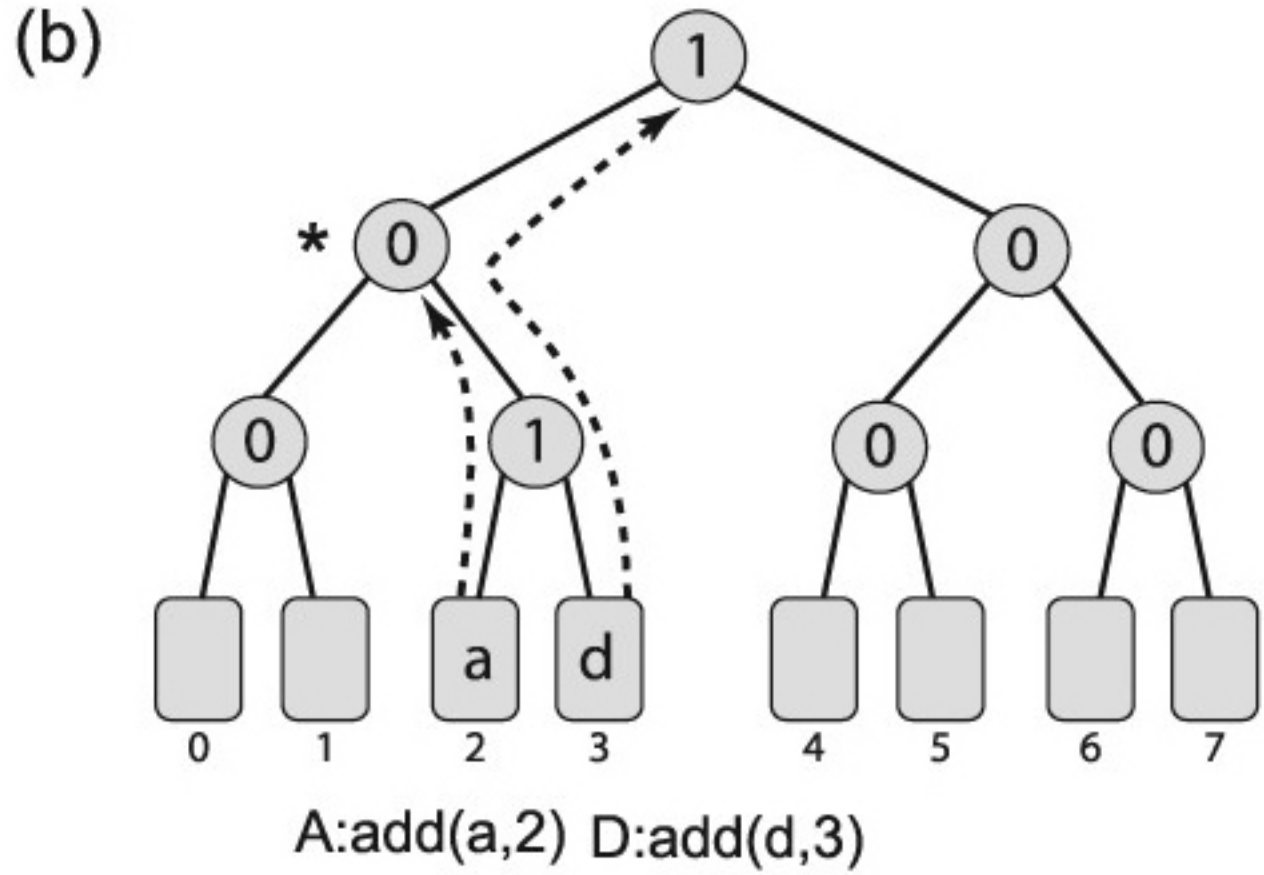

FIGURE 15.5B The SimpleTree priority queue is a tree of bounded counters. Items reside in bins at the leaves. Internal nodes hold the number of items in the subtree rooted at the node's left child. In part (a), threads *A* and *D* add items by traversing up the tree, incrementing the counters in the nodes when they ascend from the left. Thread *B* follows the counters down the tree, descending left if the counter had a nonzero value (we do not show the effect of *B*'s decrements). Parts (b)– (d) show a sequence in which concurrent threads *A* and *B* meet at the node marked by a star. In part (b), thread *D* adds *d*, and then *A* adds *a* and ascends to the starred node, incrementing a counter along the way. In part (c), *B* traverses down the tree, decrementing counters to zero and popping *a*. In part (d), *A* continues its ascent, incrementing the counter at the root even though *B* already removed any trace of *a* from the starred node down. Nevertheless, all is well, because the nonzero root counter correctly leads *C* to item *d*, the item with the highest priority.