Algorithm Design

M. T. Goodrich and R. Tamassia

John Wiley & Sons

Solution of Exercise C-2.2

Name the two stacks as E and D, for we will enqueue into E and dequeue from D. To implement enqueue(e), simply call E.push(e). To implement dequeue(), simply call D.pop(), provided that D is not empty. If D is empty, iteratively pop every element from E and push it onto D, until E is empty, and then call D.pop(). For the amortized analysis, charge \$2 to each enqueue, using \$1 to do the push into E. Imagine that we store the extra cyber-dollar with the element just pushed. We use the cyber-dollar associated with an element when we move it from E to D. This is sufficient, since we never move elements back from D to E. Finally, we charge \$1 for each dequeue to pay for the push from D (to remove the element returned). The total charges for n operations is $O(n)$; hence, each operation runs in $O(1)$ amortized time.