

**7.2-6 ★**

Argue that for any constant  $0 < \alpha \leq 1/2$ , the probability is approximately  $1 - 2\alpha$  that on a random input array, PARTITION produces a split more balanced than  $1 - \alpha$  to  $\alpha$ .

---

**7.3 A randomized version of quicksort**

In exploring the average-case behavior of quicksort, we have made an assumption that all permutations of the input numbers are equally likely. In an engineering situation, however, we cannot always expect this assumption to hold. (See Exercise 7.2-4.) As we saw in Section 5.3, we can sometimes add randomization to an algorithm in order to obtain good expected performance over all inputs. Many people regard the resulting randomized version of quicksort as the sorting algorithm of choice for large enough inputs.

In Section 5.3, we randomized our algorithm by explicitly permuting the input. We could do so for quicksort also, but a different randomization technique, called *random sampling*, yields a simpler analysis. Instead of always using  $A[r]$  as the pivot, we will select a randomly chosen element from the subarray  $A[p \dots r]$ . We do so by first exchanging element  $A[r]$  with an element chosen at random from  $A[p \dots r]$ . By randomly sampling the range  $p, \dots, r$ , we ensure that the pivot element  $x = A[r]$  is equally likely to be any of the  $r - p + 1$  elements in the subarray. Because we randomly choose the pivot element, we expect the split of the input array to be reasonably well balanced on average.

The changes to PARTITION and QUICKSORT are small. In the new partition procedure, we simply implement the swap before actually partitioning:

RANDOMIZED-PARTITION( $A, p, r$ )

```

1   $i = \text{RANDOM}(p, r)$ 
2  exchange  $A[r]$  with  $A[i]$ 
3  return PARTITION( $A, p, r$ )

```

The new quicksort calls RANDOMIZED-PARTITION in place of PARTITION:

RANDOMIZED-QUICKSORT( $A, p, r$ )

```

1  if  $p < r$ 
2       $q = \text{RANDOMIZED-PARTITION}(A, p, r)$ 
3      RANDOMIZED-QUICKSORT( $A, p, q - 1$ )
4      RANDOMIZED-QUICKSORT( $A, q + 1, r$ )

```

We analyze this algorithm in the next section.