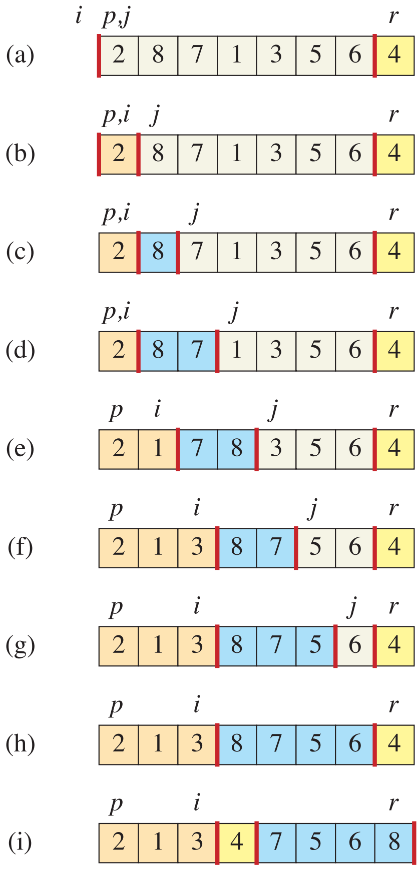

Figure 7.1 The operation of PARTITION on a sample array. Array entry $A[r]$ becomes the pivot element x. Tan array elements all belong to the low side of the partition, with values at most x. Blue elements belong to the high side, with values greater than x. White elements have not yet been put into either side of the partition, and the yellow element is the pivot x. **(a)** The initial array and variable settings. None of the elements have been placed into either side of the partition. **(b)** The value 2 is "swapped with itself" and put into the low side. **(c)–(d)** The values 8 and 7 are placed into to high side. **(e)** The values 1 and 8 are swapped, and the low side grows. **(f)** The values 3 and 7 are swapped, and the low side grows. **(g)–(h)** The high side of the partition grows to include 5 and 6, and the loop terminates. **(i)** Line 7 swaps the pivot element so that it lies between the two sides of the partition, and line 8 returns the pivot's new index.

The final two lines of PARTITION finish up by swapping the pivot with the leftmost element greater than x, thereby moving the pivot into its correct place in the partitioned array, and then returning the pivot's new index. The output of PARTITION now satisfies the specifications given for the divide step. In fact, it satisfies a slightly stronger condition: after line 3 of QUICKSORT, $A[q]$ is strictly less than every element of $A[q+1:r]$.