

Chapter 15

Searching, Sorting, and Recursive Algorithms

OBJECTIVES

After you have read and studied this chapter, you should be able to

- Write recursive algorithms for mathematical functions and non-numerical operations.
- Perform linear and binary search algorithms on small arrays.
- Determine whether linear or binary search is more effective for a given situation.
- Perform selection and bubble sort algorithms.
- Describe the recursive quicksort algorithm and explain how its performance is better than the other two sorting algorithms.

FIGURE 15.1 Successful and unsuccessful searches.

Unsuccessful Search: `search(45)` → `NOT_FOUND (-1)`

Successful Search: `search(12)` → 4

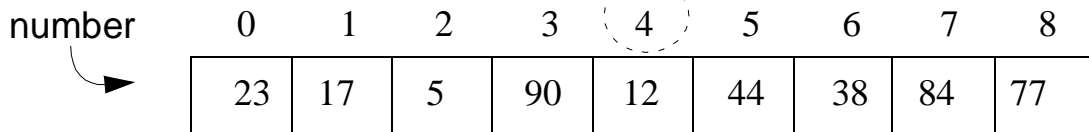
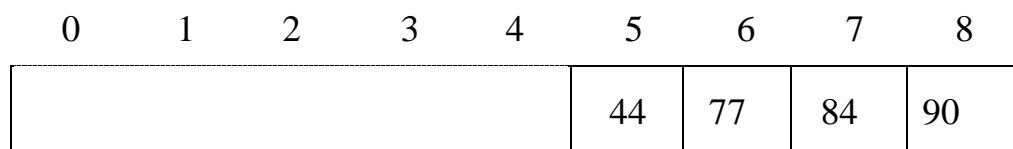
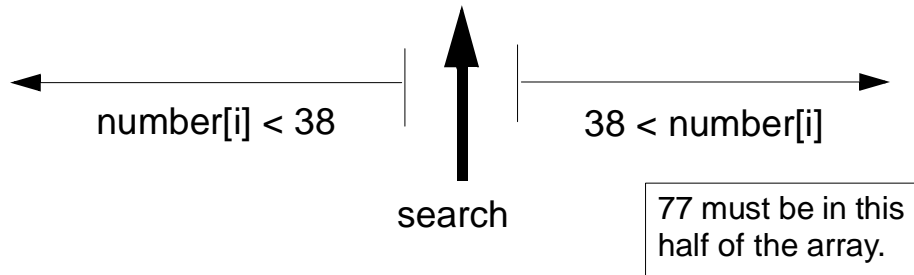
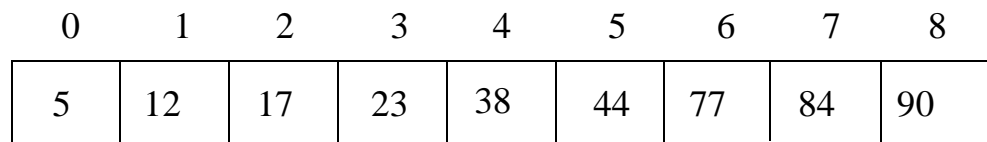


FIGURE 15.2 Effect of one comparison in binary search.

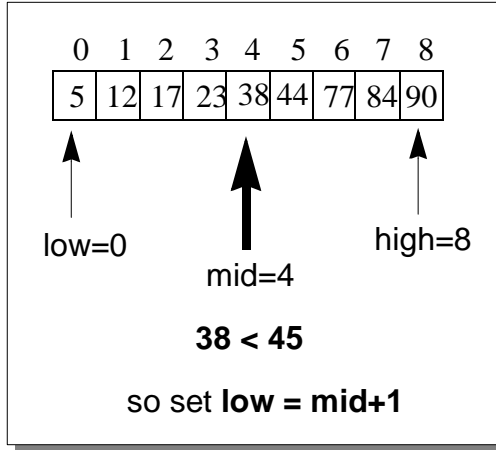


No need to consider the left half anymore.

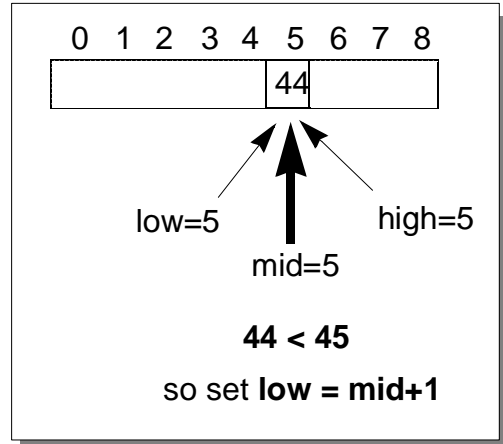
FIGURE 15.3 How the unsuccessful search is terminated in the binary search routine.

Suppose we search for **45**

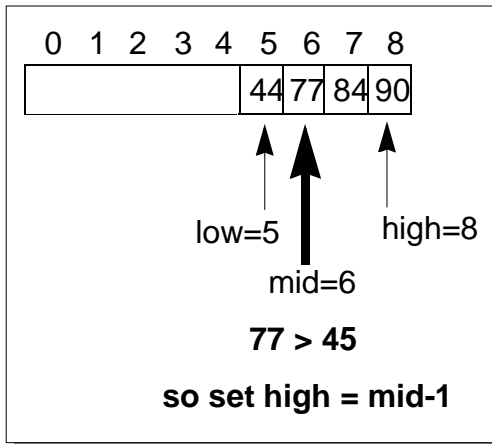
1



3



2



4

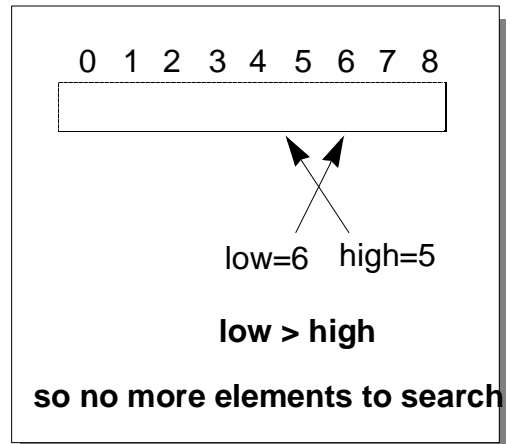


FIGURE 15.4 Human sorting algorithm after three numbers are moved to the sorted list.

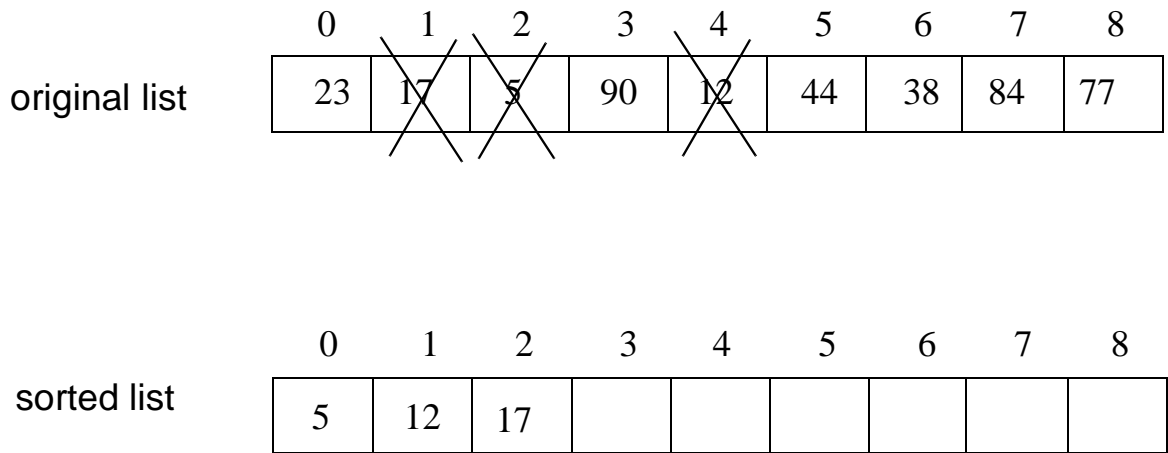


FIGURE 15.5 Effect of executing the first pass in the selection sort.

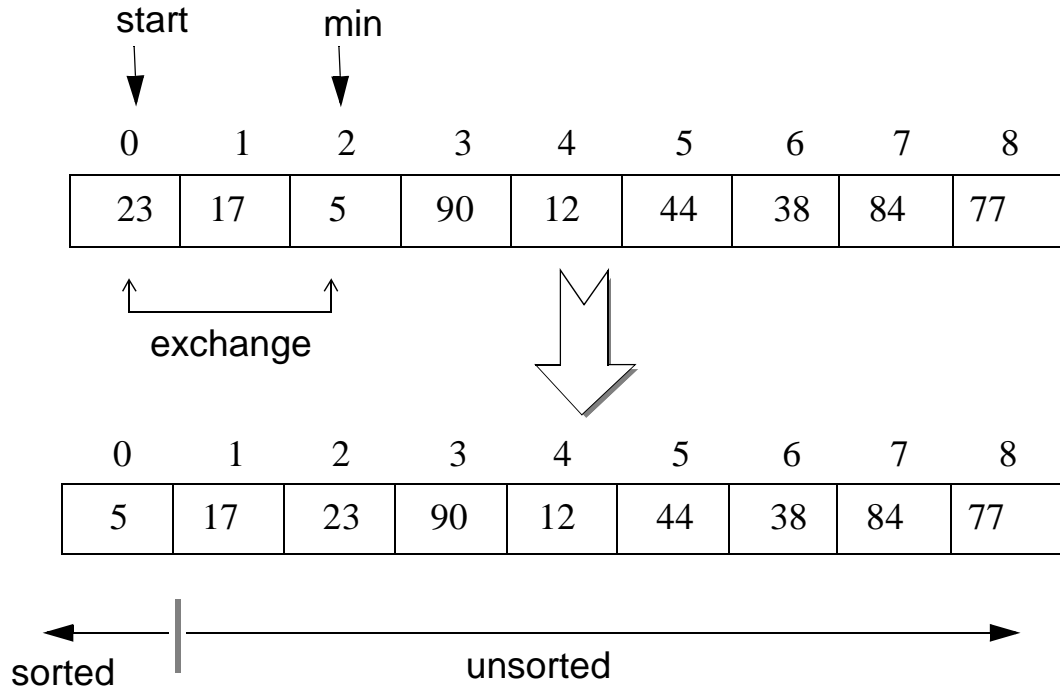
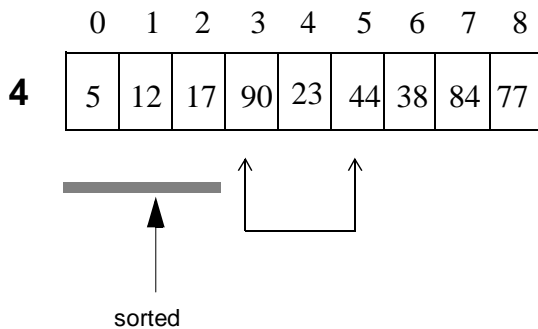
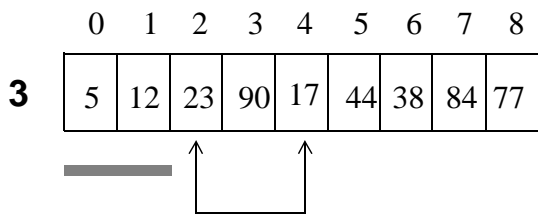
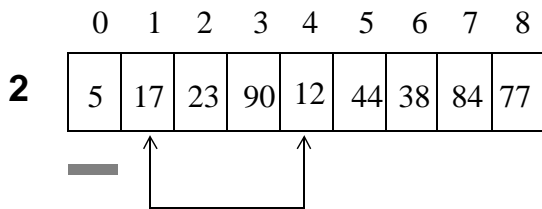
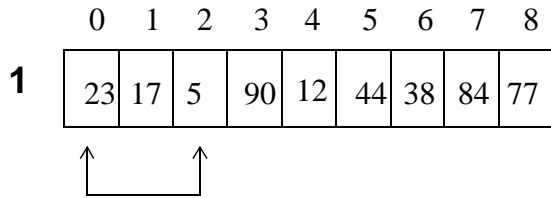


FIGURE 15.6 Eight passes to sort the sample array of nine elements.

pass



pass

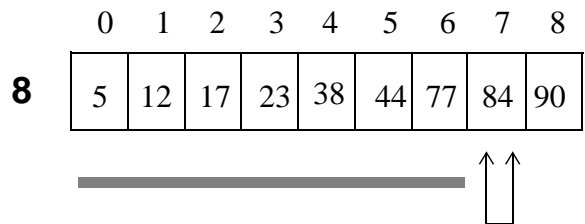
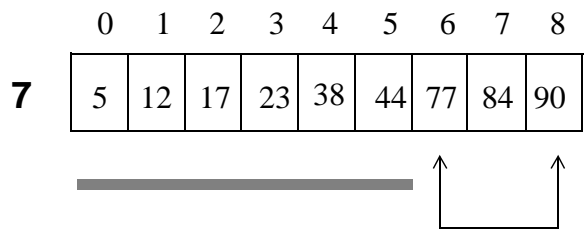
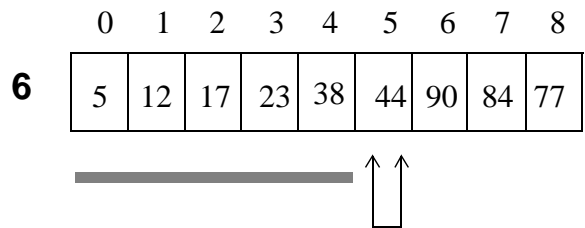
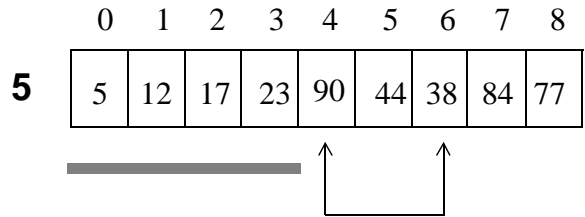


FIGURE 15.7 Effect of executing the first pass in the bubble sort.

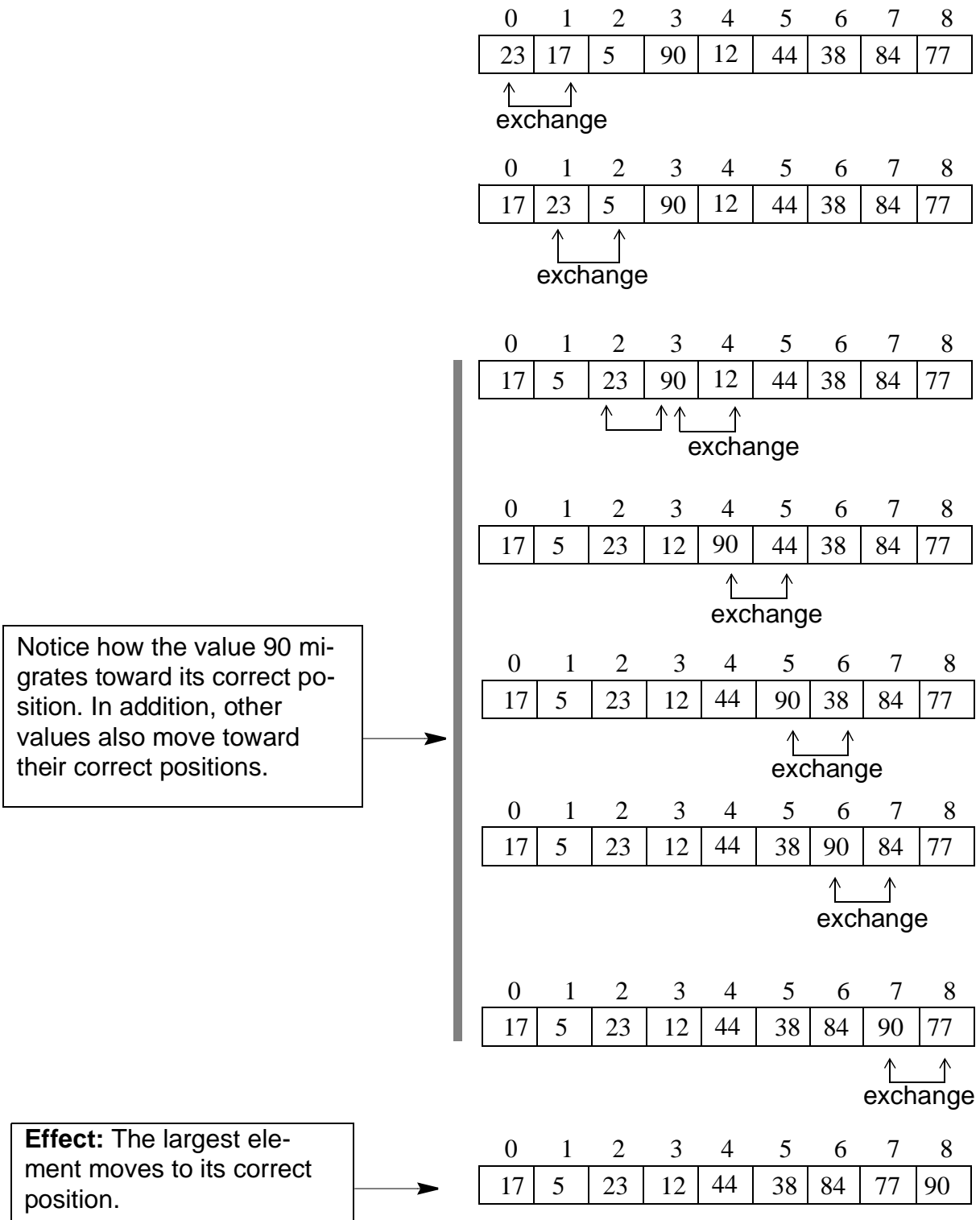


FIGURE 15.8 How to generate all anagrams of a word using recursion.

Find all anagrams of a word **H** **A** **L** **O**

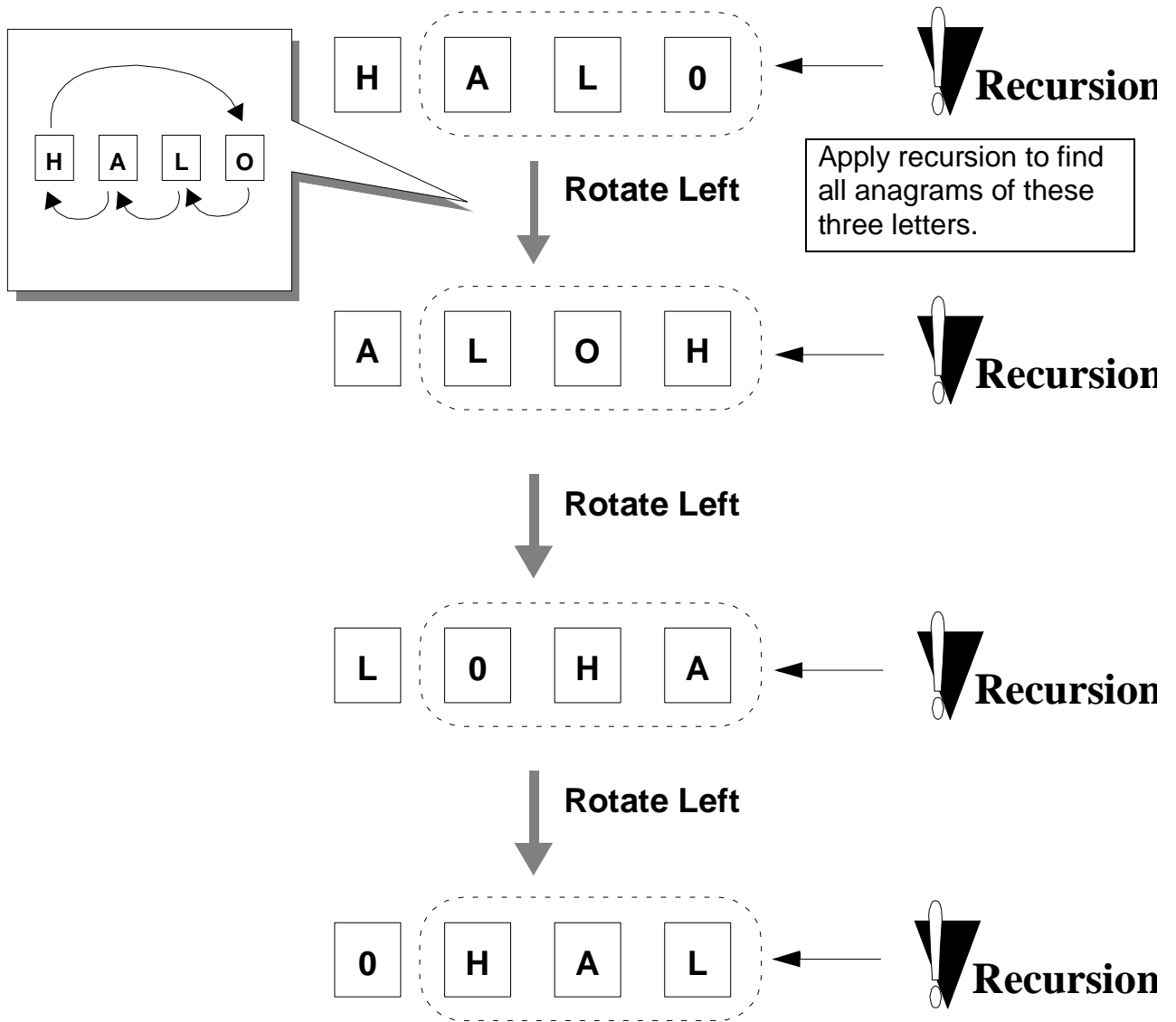


FIGURE 15.9 Tower of Hanoi with $N = 4$ disks.

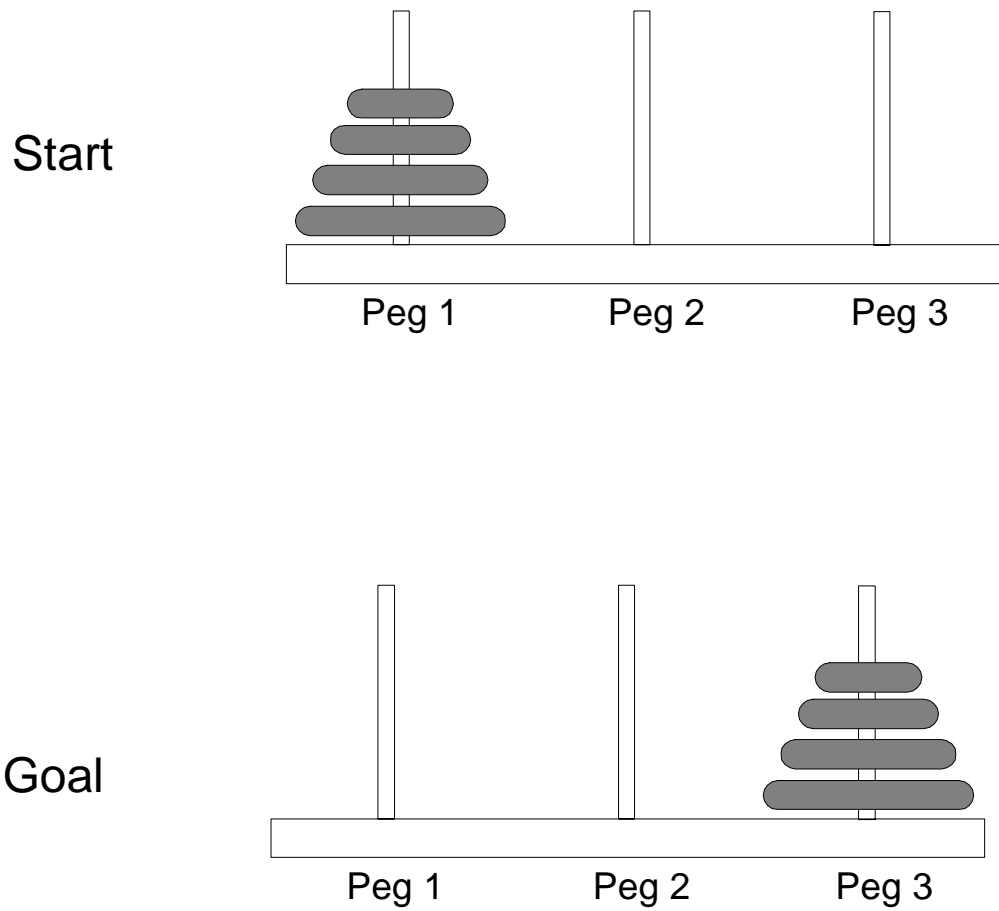


FIGURE 15.10 Recursive solution to the Tower of Hanoi puzzle.

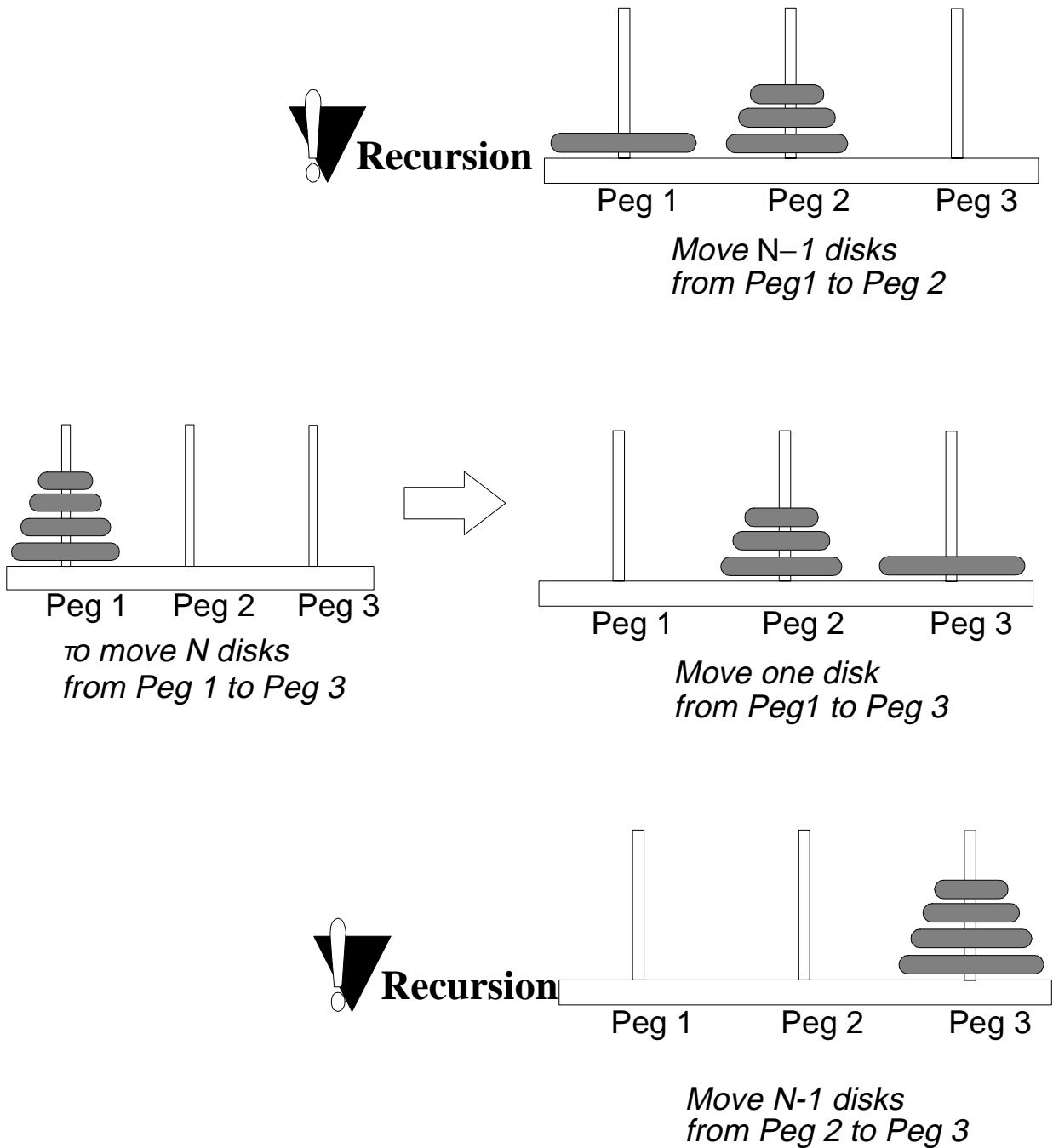


FIGURE 15.11 The core idea of the quicksort algorithm.

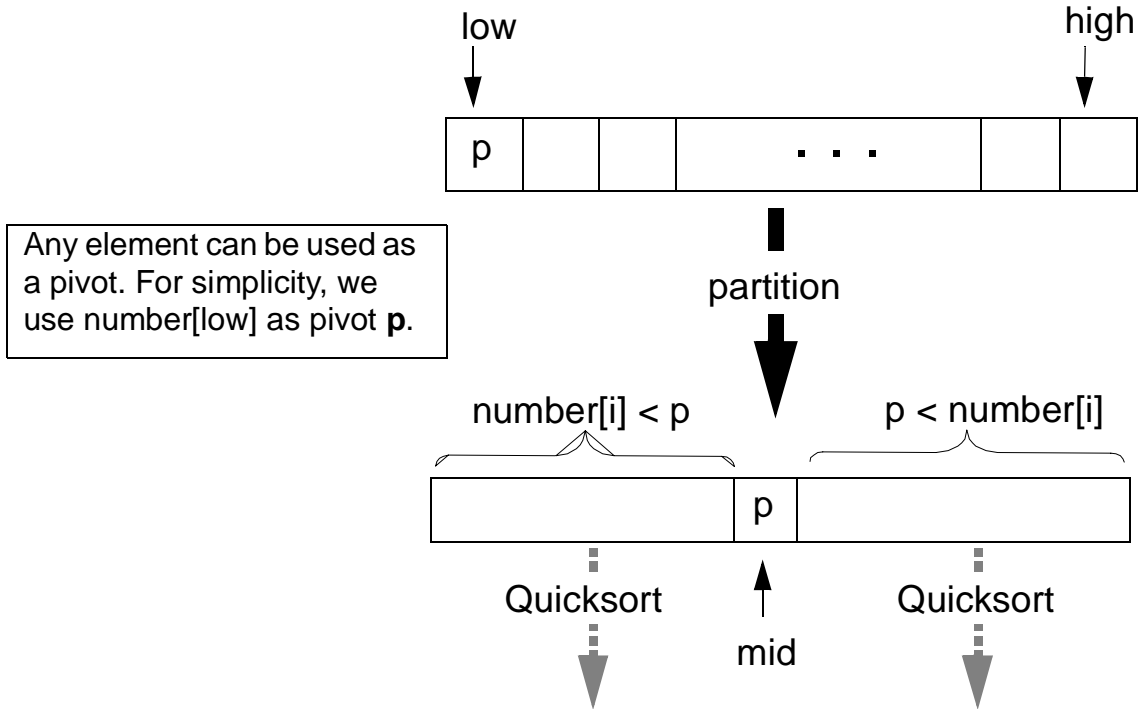


FIGURE 15.12 Result of partitioning using 23 as a pivot.

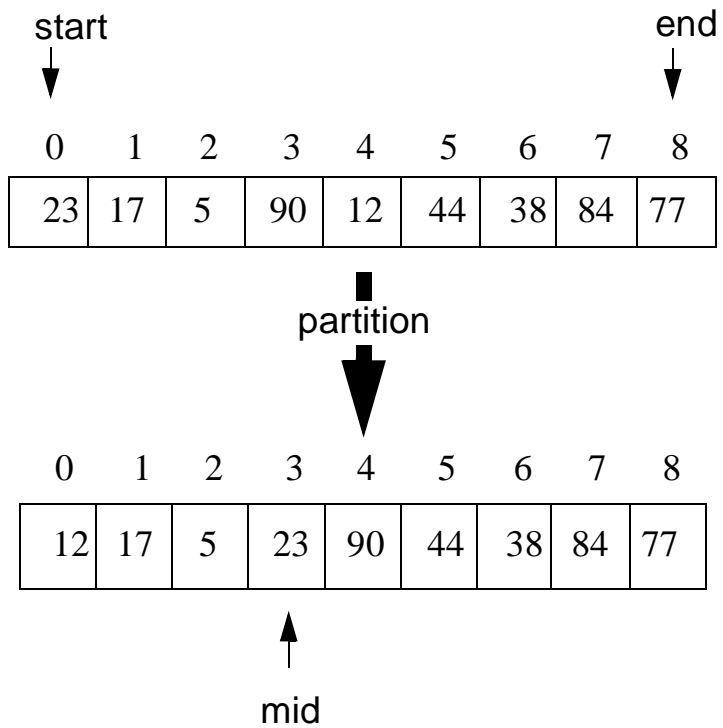


FIGURE 15.13 Details of one partitioning.

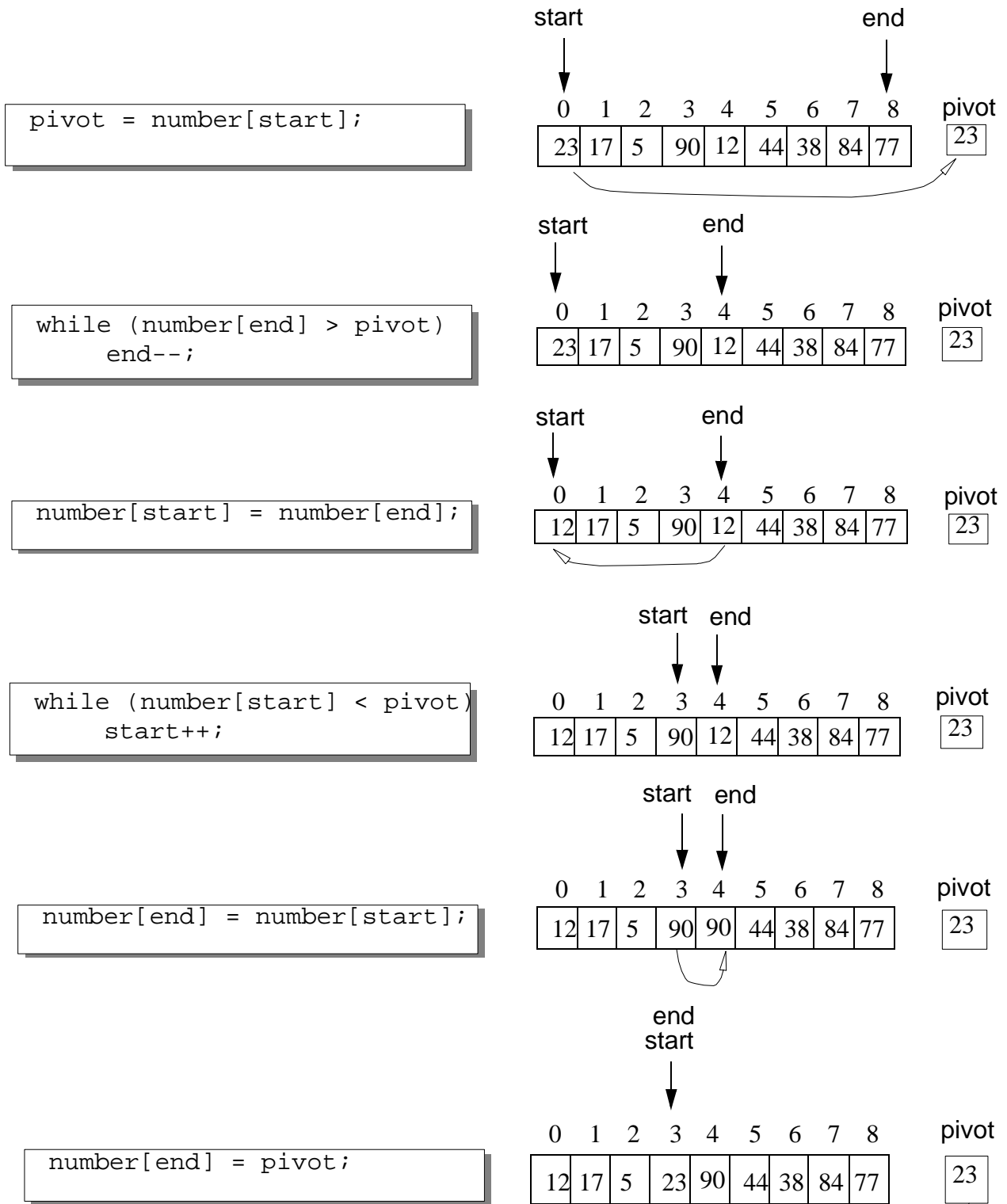


FIGURE 15.14 A hierarchy of partitioning an array into smaller and smaller arrays in the quicksort.

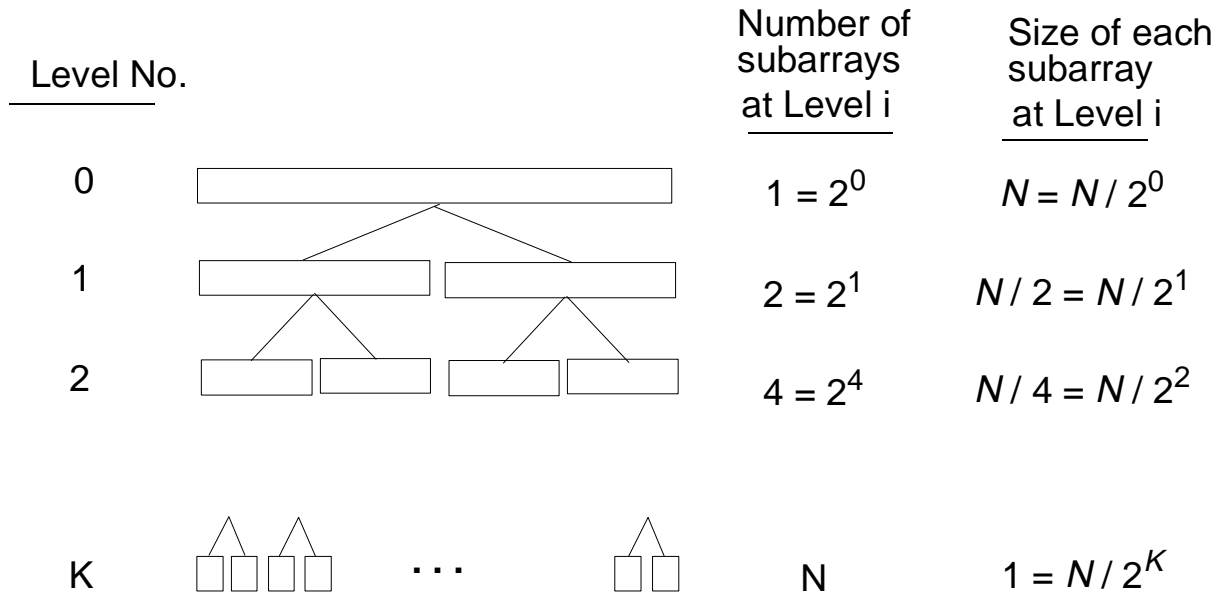


FIGURE 15.15 Recursive calls to compute **fibonacci(5)**.

