Practice Exercises

1. Show how the values in the following array would be arranged immediately before the execution of the function Merge() in the original call to MergeSort().

43	7	10	23	18	4	19	5	66	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

2. A sorting function is called to sort a list of 100 integers. If all the values are zero, what would the execution requirements (in term of Big-O notation) be if the sort used was:
   a. Quick sort, with the first element used as the split/partition value?
   b. Bubble sort?
   c. Selection sort?
   d. Heap sort?
   e. Insertion sort?
   f. Merge sort?

3. A merge sort is used to sort an array of 1000 test scores in descending order. Which of the following statements is true?
   a. The sort is fastest if the original test scores are sorted from the smallest to largest.
   b. The sort is fastest if the original test scores are in completely random order.
   c. The sort is fastest if the original test scores are sorted from the largest to smallest.
   d. The sort is the same, no matter what the order of the original elements.

4. A list is sorted from smallest to largest when a sort algorithm is called. Which of the following sorts would take the longest time to execute, and which would take the shortest time?
   a. Quick sort, with the first element used as the split/partition value?
   b. Bubble sort?
   c. Selection sort?
   d. Heap sort?
   e. Insertion sort?
   f. Merge sort?

5. Answer the following questions:
   a. In what cases, if any, is the bubble sort $O(N^2)$?
   b. In what cases, if any, is the selection sort $O(Log_2N)$?
   c. In what cases, if any, is quick sort $O(N^2)$?

6. A very large array of elements is to be sorted. The program will be run on a personal computer with limited memory. Which sort would be a better choice: a heap sort or a merge sort? Why?

7. Select the correct answers: Reordering an array of pointers to list elements, rather than sorting the elements themselves, is a good idea when:
    a. The number of elements is very large.
    b. The individual elements are large in size.
    c. The sort is recursive.
    d. There are multiple keys on which to sort the elements.

8. Explain why the heap sort algorithm is NOT stable?