More improvement is possible (as you will see in Exercise 8.18). Some of these improvements have theoretical backing, but no known sequence markedly improves the program shown in Figure 8.4.

The performance of Shellsort is quite acceptable in practice, even for N in the tens of thousands. The simplicity of the code makes it the algorithm of choice for sorting up to moderately large input. It is also a fine example of a very simple algorithm with an extremely complex analysis.

Shellsort is a good choice for moderate amounts of input.

8.5 Mergesort

Recall from Section 7.5 that recursion can be used to develop subquadratic algorithms. Specifically, a divide-and-conquer algorithm in which two half-size problems are solved recursively with an $O(N)$ overhead results in an $O(N \log N)$ algorithm. Mergesort is such an algorithm. It offers a better bound, at least theoretically, than the bounds claimed for Shellsort.

Mergesort uses divide-and-conquer to obtain an $O(N \log N)$ running time.

The mergesort algorithm involves three steps:

1. If the number of items to sort is 0 or 1, return.
2. Recursively sort the first and second halves separately.
3. Merge the two sorted halves into a sorted group.

To claim an $O(N \log N)$ algorithm, we need to show only that the merging of two sorted groups can be performed in linear time. This section shows how to merge two input arrays A and B, with the result placed in a third array C. We then provide a simple implementation of mergesort. The merge routine is the cornerstone of most external sorting algorithms. This is demonstrated in Section 20.6.

Merging of sorted arrays can be done in linear time.

8.5.1 Linear-time Merging of Sorted Arrays

The basic merge algorithm takes two input arrays A and B, output array C, and three counters *Actr*, *Bctr*, and *Cctr*, which are initially set to the beginning of their respective arrays. The smaller of *A[Actr]* and *B[Bctr]* is copied to the next entry in C, and the appropriate counters are advanced. When either input array is exhausted, the rest of the other array is copied to C.

An example of how the merge routine works is provided for the following input: