Ch 5.4 Recursive Algorithms (R)

An algorithm is called *recursive* if it solves a problem by reducing it to an instance of the same problem with smaller input.

Can use either mathematical induction or strong induction to prove its correctness.

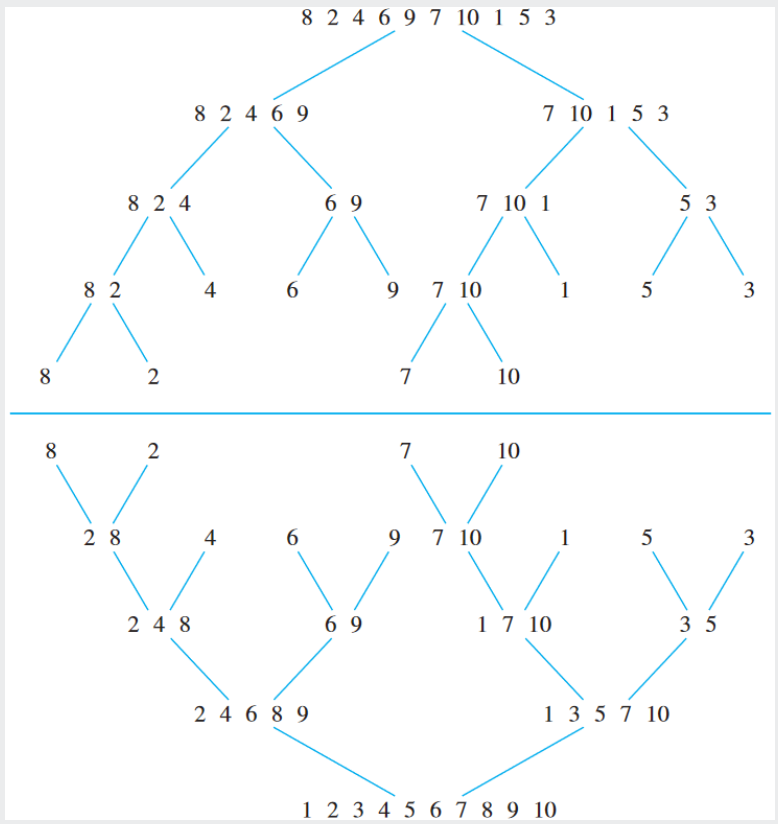
Iterative: start with the value of the function at one or more integers, the base cases, and successively apply the recursive definition to find the values of the function at successive larger integers.

Often use for the evaluation of a recursively defined sequence that requires much less computation.

▼ Example 9

Use the merge sort to put the terms of the list 8, 2, 4, 6, 9, 7, 10, 1, 5, 3 in increasing order.

1. Split list into two successively until it becomes individual elements
2. Merge by pairs successively in increasing order



ALGORITHM 9 A Recursive Merge Sort.

procedure *mergesort*($L = a_1, \ldots, a_n$)
if $n > 1$ **then**
    $m := \lfloor n/2 \rfloor$
    $L_1 := a_1, a_2, \ldots, a_m$
    $L_2 := a_{m+1}, a_{m+2}, \ldots, a_n$
    $L := merge(mergesort(L_1), mergesort(L_2))$
{L is now sorted into elements in nondecreasing order}

Two sorted lists with m elements and n elements can be merged into a sorted list using no more than $m + n - 1$ comparisons.

The number of comparisons needed to merge sort a list with n elements is $O(n \log n)$.