

# Divide and Conquer Algorithm

Authored by  
**mohammad looti**

June 5, 2026

## RECOMMENDED CITATION

mohammad looti (2026). *Divide and Conquer Algorithm*. PSYCHOLOGICAL SCALES.  
Retrieved from <https://scales.arabpsychology.com/?p=38348>

In computer science, divide and conquer (D&C) is an important algorithm design paradigm based on multi-branched recursion. A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.

This technique is the basis of efficient algorithms for all kinds of problems, such as sorting (e.g., quicksort, merge sort), multiplying large numbers (e.g. Karatsuba), syntactic analysis (e.g., top-down parsers), and computing the discrete Fourier transform (FFTs).

On the other hand, the ability to understand and design D&C algorithms is a skill that takes time to master. As when proving a theorem by induction, it is often necessary to replace the original problem by a more general or complicated problem in order to get the recursion going, and there is no systematic method for finding the proper generalization.

The name "divide and conquer" is sometimes applied also to algorithms that reduce each problem to only one subproblem, such as the binary search algorithm for finding a record in a sorted list (or its analog in numerical computing, the bisection algorithm for root finding). These algorithms can be implemented more efficiently than general divide-and-conquer algorithms; in particular, if they use tail recursion, they can be converted into simple loops. Under this broad definition, however, every algorithm that uses recursion or loops could be regarded as a "divide and conquer algorithm". Therefore, some authors consider that the name "divide and conquer" should be used only when each problem may generate two or more subproblems. The name decrease and conquer has been proposed instead for the single-subproblem class.