Data Structures - Divide and Conquer

To understand the divide and conquer design strategy of algorithms, let us use a simple real world example. Consider an instance where we need to brush a type C curly hair and remove all the knots from it. To do that, the first step is to section the hair in smaller strands to make the combing easier than combing the hair altogether. The same technique is applied on algorithms.

Divide and conquer approach breaks down a problem into multiple sub-problems recursively until it cannot be divided further. These sub-problems are solved first and the solutions are merged together to form the final solution.

The common procedure for the divide and conquer design technique is as follows –

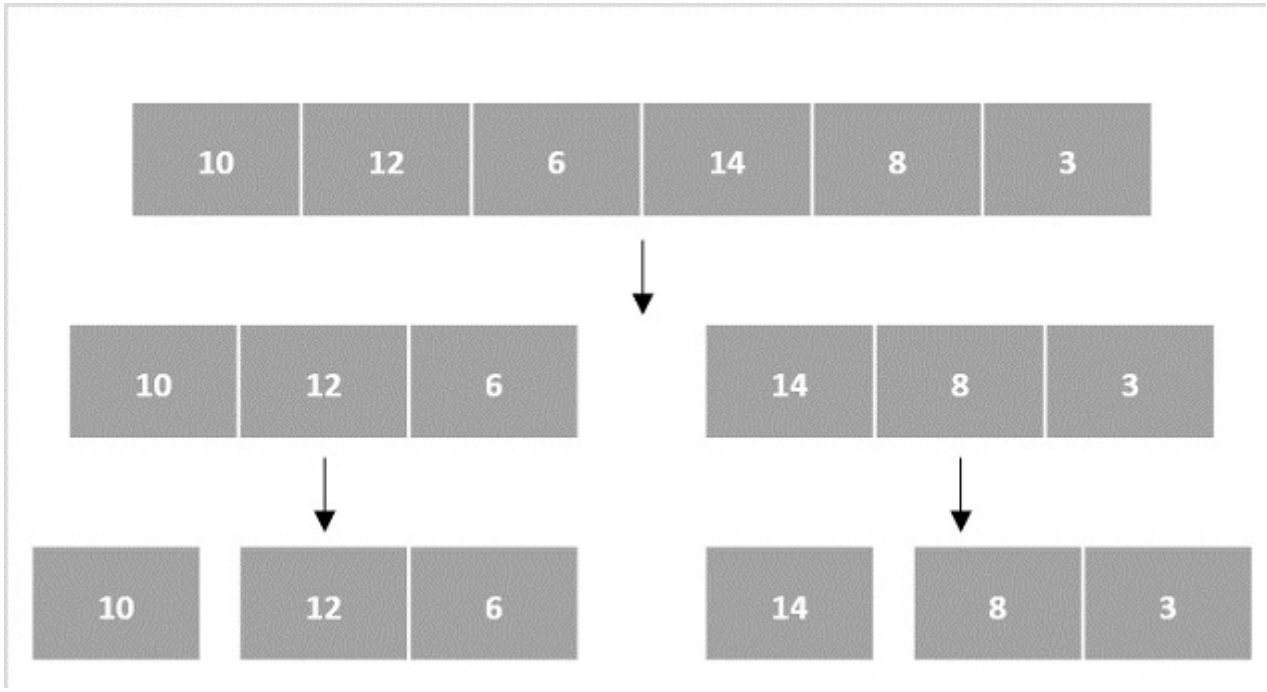
- **Divide** – We divide the original problem into multiple sub-problems until they cannot be divided further.
- **Conquer** – Then these subproblems are solved separately with the help of recursion
- **Combine** – Once solved, all the subproblems are merged/combined together to form the final solution of the original problem.

There are several ways to give input to the divide and conquer algorithm design pattern. Two major data structures used are – **arrays** and **linked lists**. Their usage is explained as

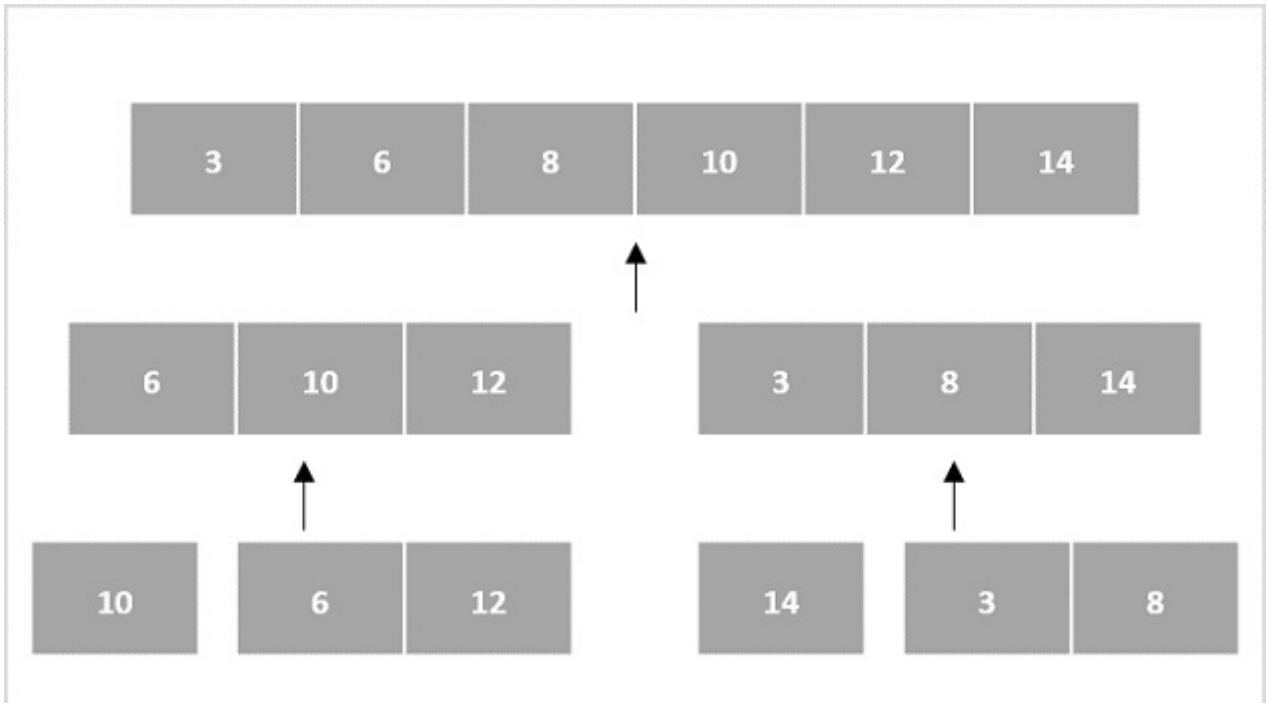
Arrays as Input

There are various ways in which various algorithms can take input such that they can be solved using the divide and conquer technique. Arrays are one of them. In algorithms that require input to be in the form of a list, like various sorting algorithms, array data structures are most commonly used.

In the input for a sorting algorithm below, the array input is divided into subproblems until they cannot be divided further.



Then, the subproblems are sorted (the conquer step) and are merged to form the solution of the original array back (the combine step).

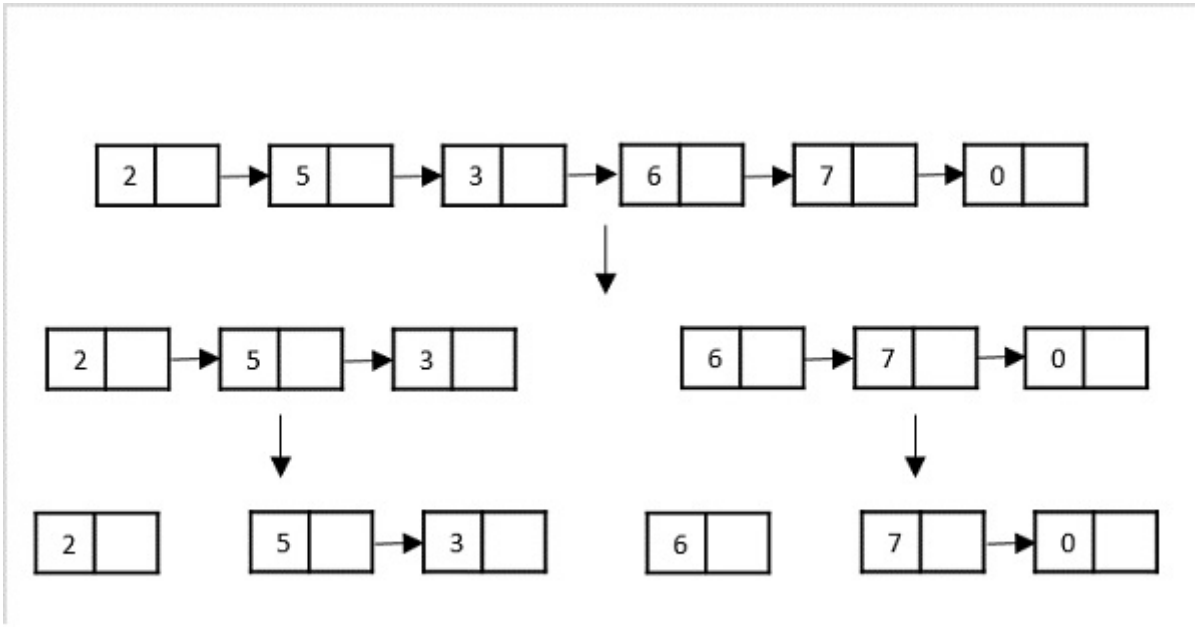


Since arrays are indexed and linear data structures, sorting algorithms most popularly use array data structures to receive input.

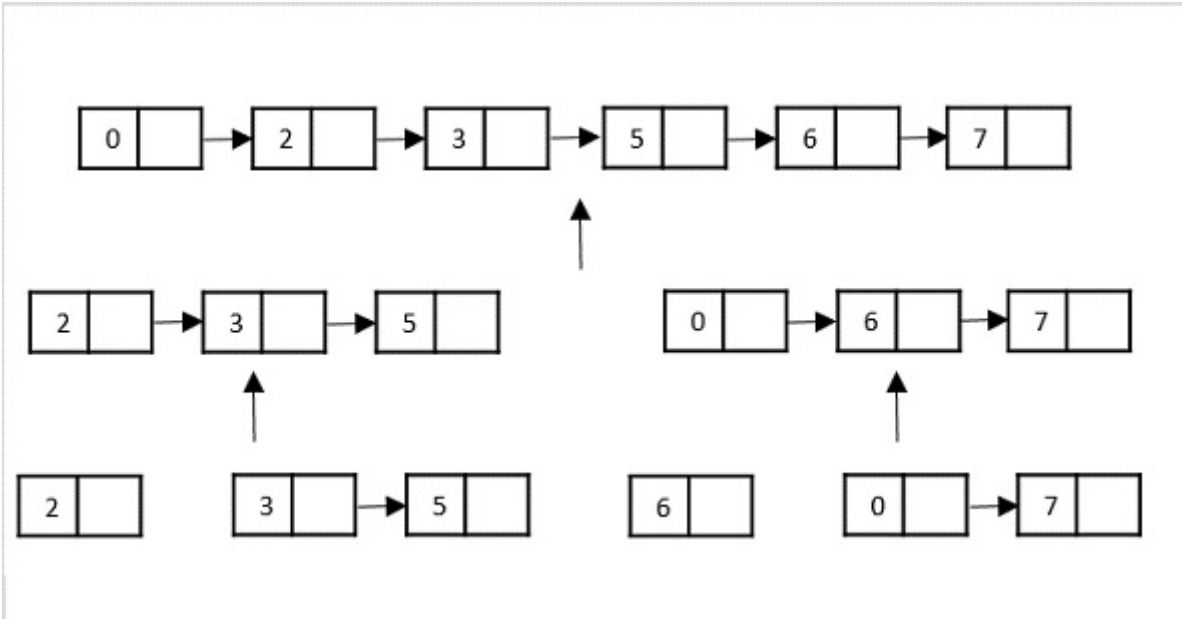
Linked Lists as Input

Another data structure that can be used to take input for divide and conquer algorithms is a linked list (for example, merge sort using linked lists). Like arrays, linked lists are also linear data structures that store data sequentially.

Consider the merge sort algorithm on linked list; following the very popular tortoise and hare algorithm, the list is divided until it cannot be divided further.



Then, the nodes in the list are sorted (conquered). These nodes are then combined (or merged) in recursively until the final solution is achieved.



Various searching algorithms can also be performed on the linked list data structures with a slightly different technique as linked lists are not indexed linear data structures. They must be handled using the pointers available in the nodes of the list.

Examples

The following computer algorithms are based on **divide-and-conquer** programming approach –

- Merge Sort
- Quick Sort
- Binary Search
- Strassen's Matrix Multiplication

- Closest Pair

There are various ways available to solve any computer problem, but the mentioned are a good example of divide and conquer approach.
