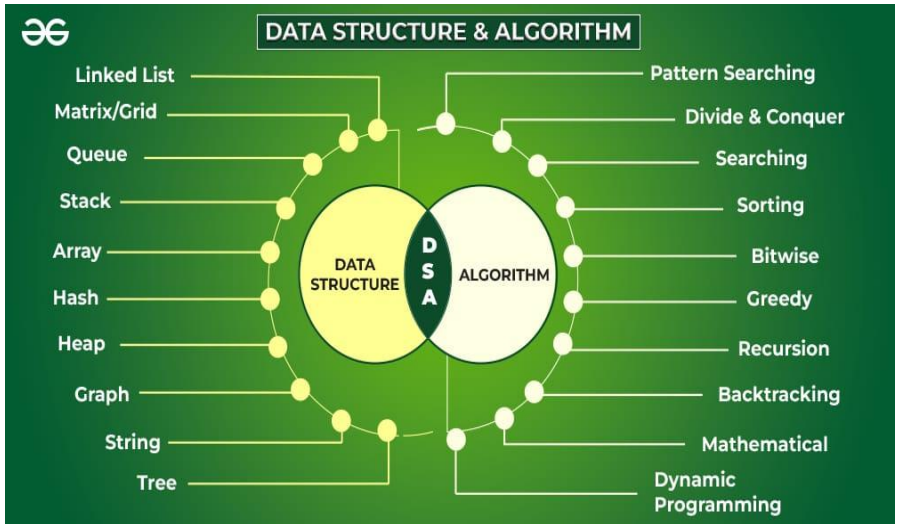
GeeksforGeeks

What is Data Structure?

CREATE BY - ATUL KUMAR (LINKEDIN)

A data structure is defined as a particular way of storing and organizing data in our devices to use the data efficiently and effectively. The main idea behind using data structures is to minimize the time and space complexities. An efficient data structure takes minimum memory space and requires minimum time to execute the data.



What is Algorithm?

Algorithm is defined as a process or set of well-defined instructions that are typically used to solve a particular group of problems or perform a specific type of calculation. To explain in simpler terms, it is a set of operations performed in a step-by-step manner to execute a task.

How to start learning DSA?

The first and foremost thing is dividing the total procedure into little pieces which need to be done sequentially.

The complete process to learn DSA from scratch can be broken into 4 parts:

- Learn about Time and Space complexities
- Learn the basics of individual Data Structures
- Learn the basics of Algorithms
- Practice Problems on DSA

Learn Algorithms

Once you have cleared the concepts of Data Structures, now its time to start your journey through the Algorithms. Based on the type of nature and usage, the Algorithms are grouped together into several categories, as shown below:

1. Searching Algorithm

Now we have learned about some linear data structures and is time to learn about some basic and most used algorithms which are hugely used in these types of data structures. One such algorithm is the searching algorithm.

Searching algorithms are used to find a specific element in an array, string, linked list, or some other data structure.

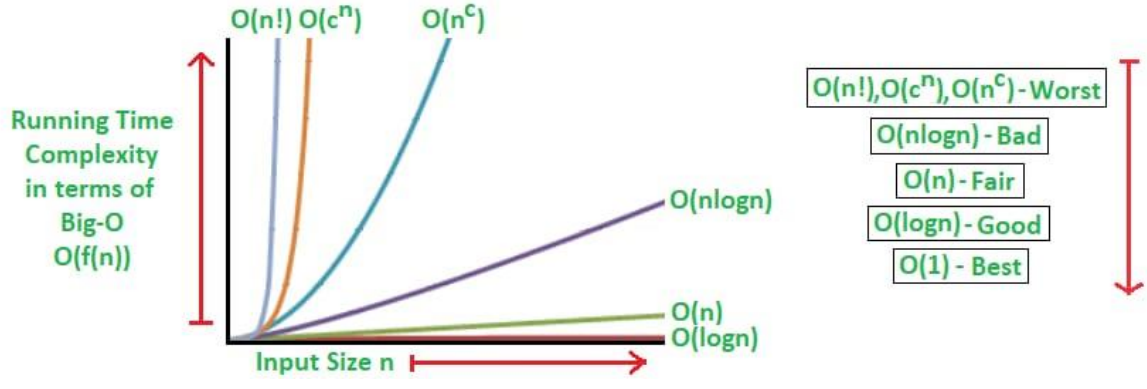
<u>The most common searching algorithms are:</u>

- Linear Search – In this searching algorithm, we check for the element iteratively from one end to the other.

- Binary Search – In this type of searching algorithm, we break the data structure into two equal parts and try to decide in which half we need to find for the element.

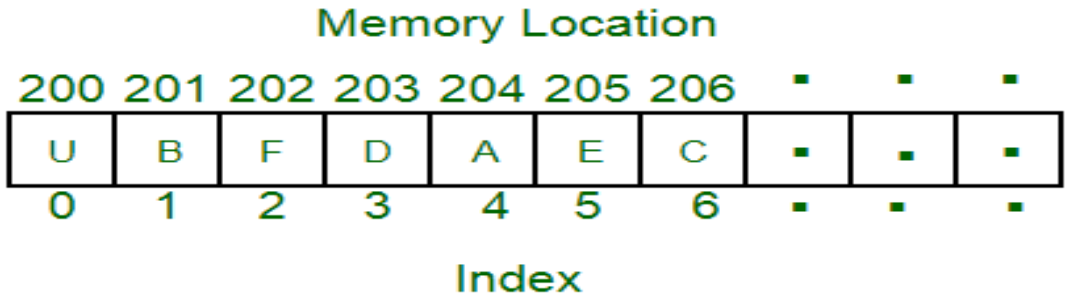
Besides these, there are other searching algorithms also like

- Jump Search
- Interpolation Search
- Exponential Search



2. Sorting Algorithm

Here is one other most used algorithm. Often we need to arrange or sort data as per a specific condition. The sorting algorithm is the one that is used in these cases. Based on conditions we can sort a set of homogeneous data in order like sorting an array in increasing or decreasing order.



Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure.

- An example to show Sorting
- An example to show Sorting

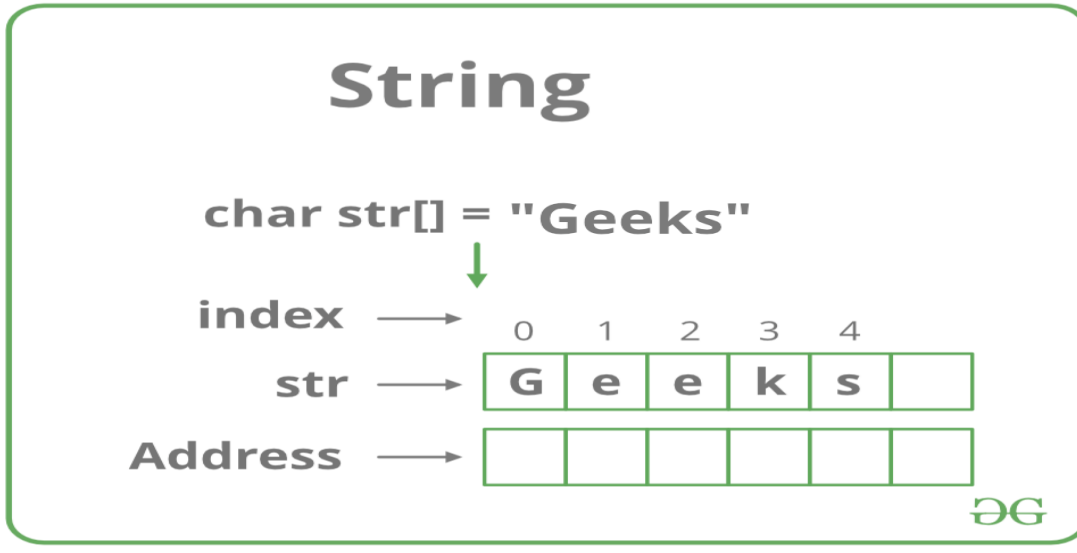
There are a lot of different types of sorting algorithms. Some widely used algorithms are:

- Bubble Sort
- Selection Sort
- Insertion Sort
- Quick Sort
- Merge Sort

There are several other sorting algorithms also and they are beneficial in different cases. You can learn about them and more in our dedicated article on Sorting algorithms.

3. Divide and Conquer Algorithm

This is one interesting and important algorithm to be learned in your path of programming. As the name suggests, it breaks the problem into parts, then solves each part and after that again merges the solved subtasks to get the actual problem solved.



Divide and Conquer is an algorithmic paradigm. A typical Divide and Conquer algorithm solves a problem using following three steps.

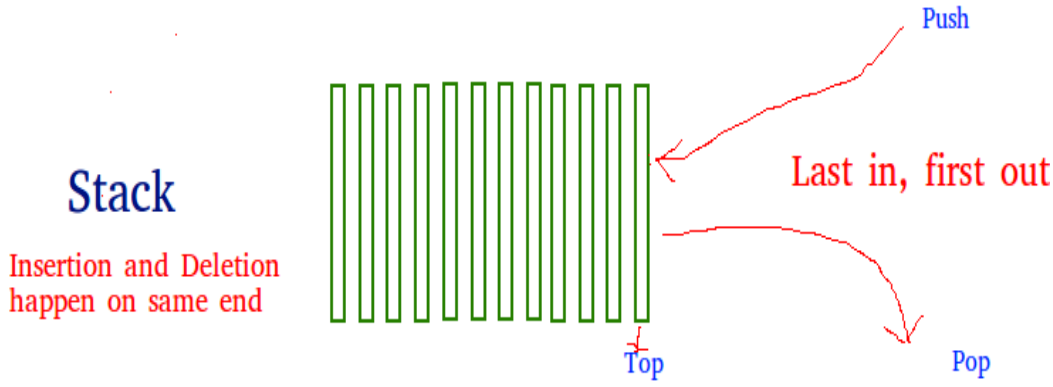
Divide: Break the given problem into subproblems of same type.

Conquer: Recursively solve these subproblems
Combine: Appropriately combine the answers

This is the primary technique mentioned in the two sorting algorithms Merge Sort and Quick Sort which are mentioned earlier. To learn more about the technique, the cases where it is used, and its implementation and solve some interesting problems, please refer to the dedicated article Divide and Conquer Algorithm.

4. Greedy Algorithms

As the name suggests, this algorithm builds up the solution one piece at a time and chooses the next piece which gives the most obvious and immediate benefit i.e., which is the most optimal choice at that moment. So the problems where choosing locally optimal also leads to the global solutions are best fit for Greedy.



For example, consider the Fractional Knapsack Problem. The local optimal strategy is to choose the item that has maximum value vs weight ratio. This strategy also leads to a globally optimal solution because we are allowed to take fractions of an item.

- Fractional Knapsack Problem
- Fractional Knapsack Problem

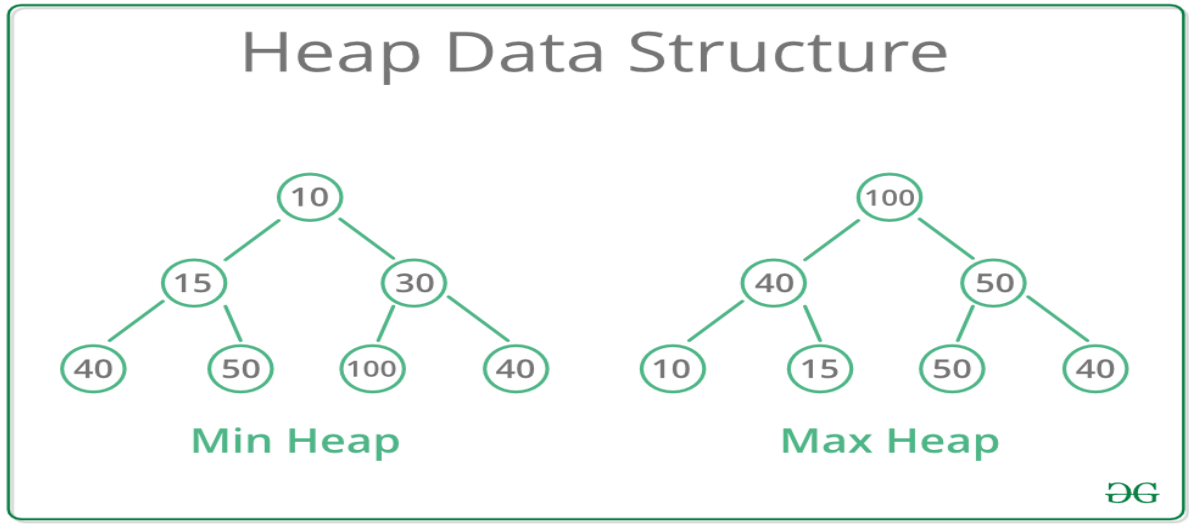
Here is how you can get started with the Greedy algorithm with the help of relevant sub-topics:

- Standard greedy algorithms
- Greedy algorithms in graphs
- Greedy Algorithms in Operating Systems
- Greedy algorithms in array
- Approximate greedy algorithms for NP-complete problems

5. Recursion

Recursion is one of the most important algorithms which uses the concept of code reusability and repeated usage of the same piece of code.

- Recursion
- Recursion



The point which makes Recursion one of the most used algorithms is that it forms the base for many other algorithms such as:

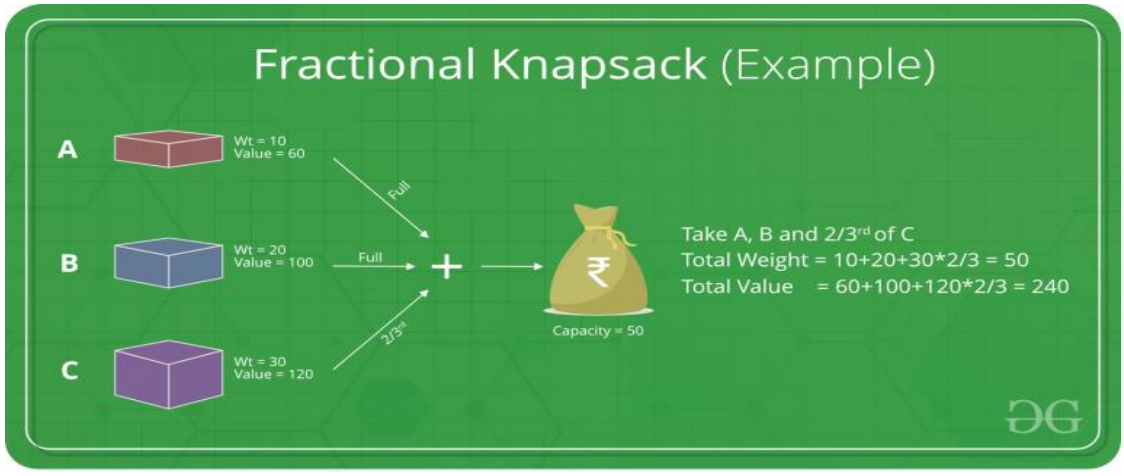
- Tree traversals
- Graph traversals
- Divide and Conquers Algorithms
- Backtracking algorithms

In Recursion, you can follow the below articles/links to get the most out of it:

- Recursion
- Recursive Functions
- Tail Recursion
- Towers of Hanoi (TOH)

6. Backtracking Algorithm

As mentioned earlier, the Backtracking algorithm is derived from the Recursion algorithm, with the option to revert if a recursive solution fails, i.e. in case a solution fails, the program traces back to the moment where it failed and builds on another solution. So basically it tries out all the possible solutions and finds the correct one.



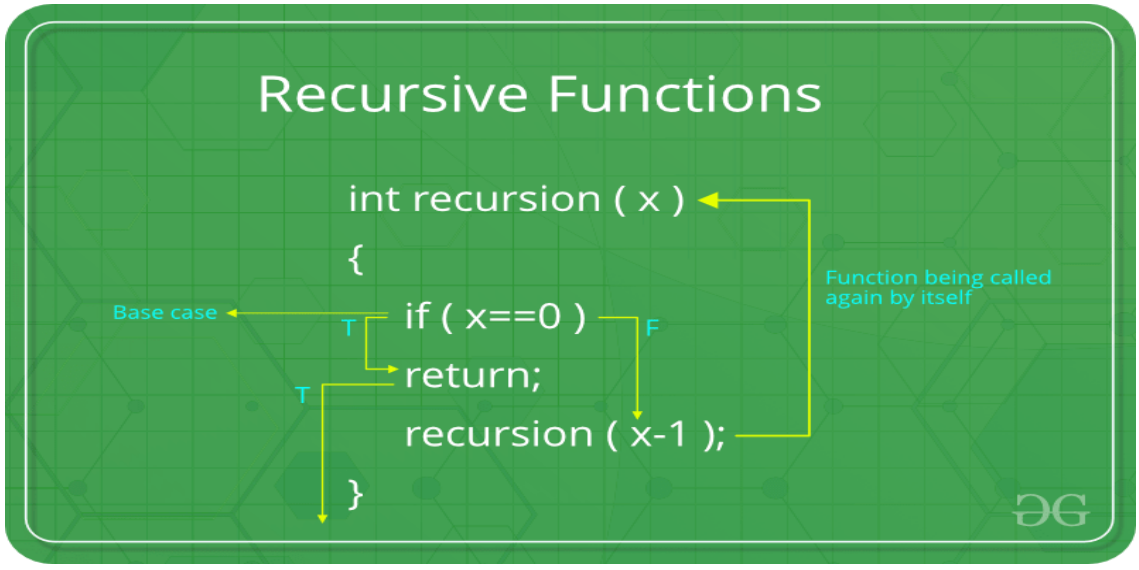
Backtracking is an algorithmic technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time

Some important and most common problems of backtracking algorithms, that you must solve before moving ahead, are:

- Knight’s tour problem
- Rat in a maze
- N-Queen problem
- Subset sum problem
- m-coloring problem
- Hamiltonian cycle
- Sudoku

7. Dynamic Programming

Another crucial algorithm is dynamic programming. Dynamic Programming is mainly an optimization over plain recursion. Wherever we see a recursive solution that has repeated calls for the same inputs, we can optimize it using Dynamic Programming.



The main concept of the Dynamic Programming algorithm is to use the previously calculated result to avoid repeated calculations of the same subtask which helps in reducing the time complexity.

To learn more about dynamic programming and practice some interesting problems related to it, refer to the following articles:

- Tabulation vs Memoization
- Optimal Substructure Property
- Overlapping Subproblems Property
- How to solve a Dynamic Programming Problem?
- Bitmasking and Dynamic Programming | Set 1
- Bitmasking and Dynamic Programming | Set-2 (TSP)
- Digit DP | Introduction

8. Pattern Searching

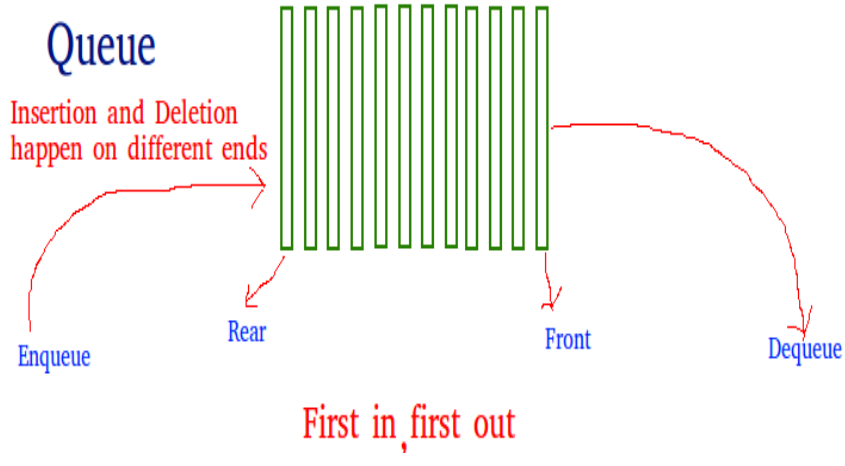
The Pattern Searching algorithms are sometimes also referred to as String Searching Algorithms and are considered as a part of the String algorithms. These algorithms are useful in the case of searching a string within another string.

Pattern Searching Algorithms

9. Mathematical Algorithms

These algorithms are designed to solve Mathematical and Number Theory problems. They requires in-depth knowledge of different mathematical subjects like

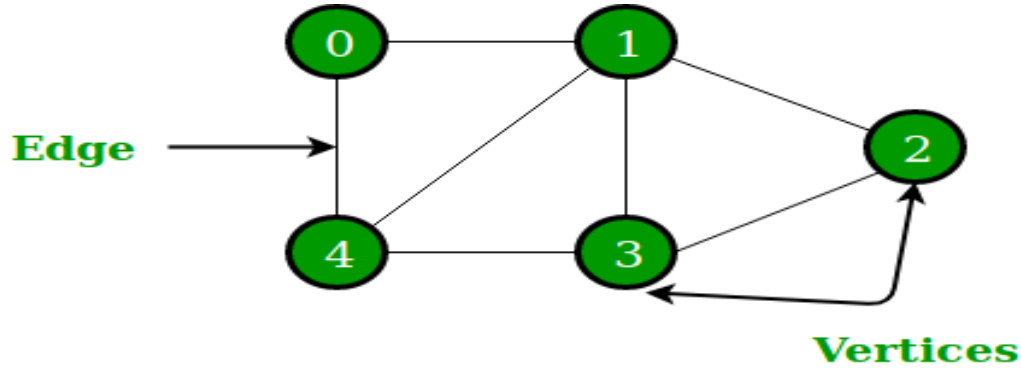
- GCD and LCM
- Prime Factorization and Divisors
- Fibonacci Numbers
- Catalan Numbers
- Modular Arithmetic
- Euler Totient Function
- nCr Computations
- Set Theory
- Factorial
- Prime numbers and Primality Tests
- Sieve Algorithms, etc.



10. Geometric Algorithms

These algorithms are designed to solve Geometric Problems. They requires in-depth knowledge of different mathematical subjects like:

- Lines
- Triangle
- Rectangle
- Square
- Circle
- 3D Objects
- Quadilateral
- Polygon & Convex Hull



For Example: Comparing Slopes of two lines, Finding Equation of a plane etc.

GeeksforGeeks

CREATE BY - ATUL KUMAR (LINKEDIN)