**Algorithm**

There are some top algorithm in data structure, every programmer should know –

1. Binary Search Algorithm
2. Sort Algorithm
3. Hashing

**Binary Search Algorithm:**

Binary search is a fast search algorithm with run-time complexity of Ο(log n). This search algorithm works on the principle of divide and conquer. For this algorithm to work properly, the data collection should be in the sorted form.

**Pseudocode:**

Procedure binary\_search

A ← sorted array

n ← size of array

x ← value to be searched

Set lowerBound = 1

Set upperBound = n

while x not found

if upperBound < lowerBound

EXIT: x does not exists.

set midPoint = lowerBound + ( upperBound - lowerBound ) / 2

if A[midPoint] < x

set lowerBound = midPoint + 1

if A[midPoint] > x

set upperBound = midPoint - 1

if A[midPoint] = x

EXIT: x found at location midPoint

end while

end procedure

**Sorting Algorithm:** A sorting algorithm is an algorithm that puts elements of a list in a certain order.

A 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.

**For example**: The below list of characters is sorted in increasing order of their ASCII values. That is, the character with lesser ASCII value will be placed first than the character with higher ASCII value.

**Hash Function:**

Hashing algorithms are functions that generate a fixed-length result (the hash, or hash value) from a given input. The hash value is a summary of the original data.

For instance, think of a paper document that you keep crumpling to a point where you aren’t even able to read its content anymore. It’s almost (in theory) impossible to restore the original input without knowing what the starting data was.

**hello**