An algorithm is a finite sequence of precise instructions for performing a computation or for solving a problem

A problem involving the location of an element in an ordered list

also known as Sequential Search, a type of algorithm that begins by comparing x to an, where the solution is ak when x=ak and 0≤k≤n. If x≠an for all n, then the solution is 0.

A search algorithm that compares elements in a list by assuming that the desired term is the middle element of a list. If not, the search is then restricted to a sublist that has endpoints based on the comparison of the previous list against the desired element to be located in the middle term.

The action of putting elements into an ordered list of increasing size

A sorting algorithm that puts a list into increasing order by successively comparing adjacent elements and swapping the two elements if they are not in increasing order.

A sorting algorithm that begins with the second element in a list of n elements. The element is then compared to the preceding elements; it is either placed before the preceding element if it does not exceed the value of the preceding element, or it is placed after the element if it does exceed the preceding element’s value.

Problems where the goal is to minimize or maximize the value of a certain parameter within the problem.

Algorithms that appear to make the “best” choice at all steps.

A well-known computer science problem that is unsolvable. It entails the impossibility of a procedure that can take a computer program as input and input to the program and determine whether the program will eventually stop when run with this input.

allows the estimation of growth of functions without needing the consideration of constant multipliers and smaller order terms

the name of the big-O symbol, named after the German mathematician Paul Landau

provides the lower bound for the size of f(x)

The notation used when providing both the upper bound and lower bound of a function f(x) relative to a reference function g(x).

The efficiency of an algorithm based on the amount of time needed based on a specified input size and the size of memory needed to implement it.

An analysis of the time required to solve a problem of a particular size

An analysis of the computer memory required of an algorithm

the largest number of operations needed to solve the given problem using this algorithm on input of specified size

The average number of operations used to solve the problem over all possible inputs of a given size

a general approach based on a particular concept that can be used to construct algorithms for solving a variety of problems

an algorithm where a problem is solved in the most straightforward manner based on the statement of the problem and the definitions of terms

When the time or space complexity of an algorithm remains the same regardless of the input

When the complexity of an algorithm relies on either the average-case or worst-case scenario

When the complexity of an algorithm relies solely on the worst-case scenario

When the time complexity of an algorithm is n log n

When the time complexity of an algorithm is Θ(n^b)

When the time complexity of an algorithm is Θ(b^n), where b>1

When the time complexity of an algorithm is Θ(n!)

a problem that is solvable using an algorithm with polynomial worst-case complexity

problems that cannot be solved using an algorithm with worst-case polynomial time complexity

problems where it can be shown that there is no algorithm that exists for solving them

problems that can be solved using an algorithm