



Learning Objectives

At the end of this chapter the students will be able to:

- Know the basics and technical perspective of algorithms.
- Understand the efficiency, time and space complexity of an algorithm.
- Develop and analyze algorithms for searching and sorting.
- Learn about dynamic programming through algorithmic approach.

4.1 Introduction to Algorithmic strategies

An algorithm is a finite set of instructions to accomplish a particular task. It is a step-by-step procedure for solving a given problem. An algorithm can be implemented in any suitable programming language.

Algorithms must have input, output and should satisfy the following characteristics such as definiteness, correctness and effectiveness. Data are maintained and manipulated effectively through data structures. Algorithms can be developed to store, manipulate and retrieve data from such data structures. Examples for data structures are arrays, structures, list, tuples, dictionary etc.

Search	To search an item in a data structure using linear and binary search.
Sort	To sort items in a certain order using the methods such as bubble sort, insertion sort, selection sort, etc.
Insert	To insert an item (s) in a data structure.
Update	To update an existing item (s) in a data structure.
Delete	To delete an existing item (s) in a data structure.

The way of defining an algorithm is called algorithmic strategy. For example to calculate factorial for the given value n then it can be done by defining the function to calculate factorial once for the iteration-1 then it can be called recursively until the number of required iteration is reached.



The word Algorithm comes from the name of a Persian author, Abu Jafar Mohammed ibn Musa al Khwarizmi (c. 825 AD(CE)), who wrote a textbook on mathematics. The word Algorithm has come to refer to a method to solve a problem.

4.1.1 Characteristics of an Algorithm

An algorithm should have the following characteristics: