2023

Victoria University of Bangladesh

58/11/A, Panthapath, Dhaka, Bangladesh

Algorithm

CSI-227

Mid Assessment

Submitted By:

ARS Nuray Alam Parash

ID # 2120180041

18th Batch, BSc in CSE

E-mail: chatokparash@gmail.com

Mobile: 01922339393

Submission Date: 19 April 2023

Submitted To:

Umme Khadiza Tithi

Lecturer

Department of Computer Science & Engineering

Victoria University of Bangladesh (VUB)

Answer to the Question No- 1 (a)

Algorithm:

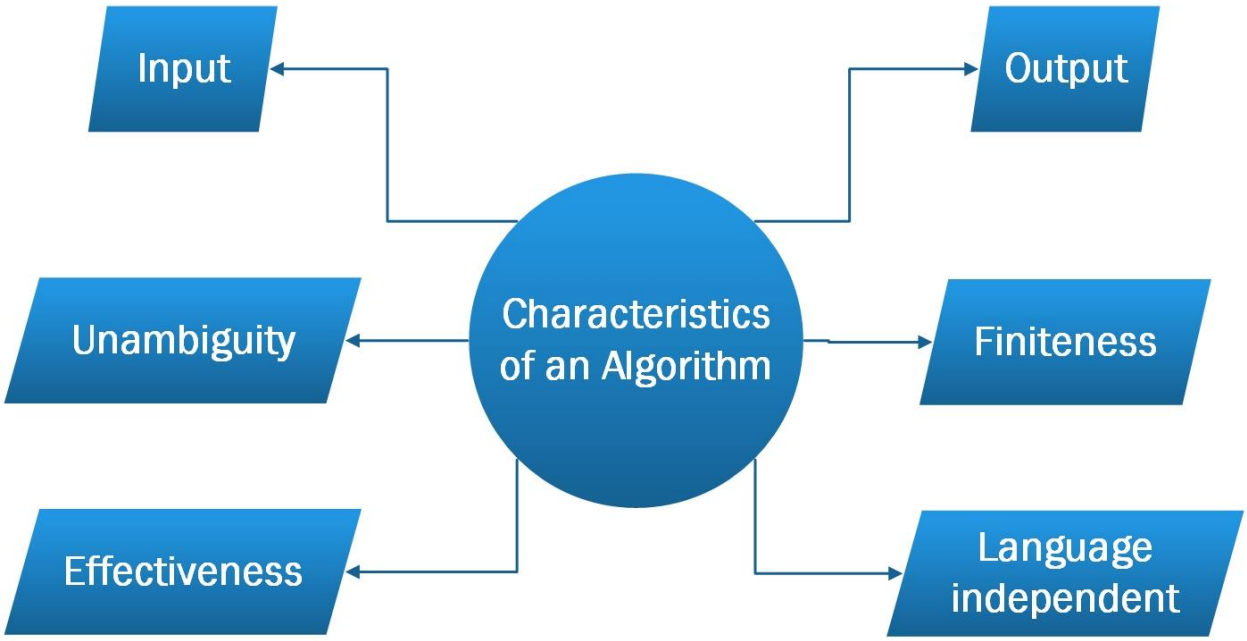
An algorithm is a process or a set of rules required to perform calculations or other problem-solving operations, especially by a computer. The formal definition of an algorithm is that it contains a finite set of instructions that are being carried out in a specific order to perform a specific task. It is not the complete program or code; it is just a solution (logic) to a problem, which can be represented as an informal description using a Flowchart or Pseudocode.

Characteristics of an Algorithm:

An algorithm has the following characteristics-

Input: An algorithm requires some input values. An algorithm can be given a value other than 0 as input.

Output: You will have one or more outcomes at the end of an algorithm.



Unambiguity: A perfect algorithm is defined as unambiguous, which means that its instructions should be clear and straightforward.

Finiteness: An algorithm must be finite. Finiteness in this context means that the algorithm should have a limited number of instructions, i.e., the instructions should be countable.

Effectiveness: Because each instruction in an algorithm affects the overall process, it should be adequate.

Language Independence: An algorithm must be language-independent, which means that its instructions can be implemented in any language and produce the same results.

Moving on in this What is an Algorithm tutorial, you will look at why you need an algorithm.

Answer to the Question No- 1 (b)

Disadvantages of Algorithms:

- Algorithms are time-consuming.
- Difficult to show Branching and Looping in Algorithms.
- Big tasks are difficult to put in Algorithms.

Advantages of Algorithms:

- It is a step-wise representation of a solution to a given problem, which makes it easy to understand.
- An algorithm uses a definite procedure.

- It is not dependent on any programming language, so it is easy to understand for anyone even without programming knowledge.
- Every step in an algorithm has its own logical sequence so it is easy to debug.
- By using the algorithm, the problem is broken down into smaller pieces or steps hence, it is easier for a programmer to convert it into an actual program.

Answer to the Question No- 1 (c)

An algorithm consists of instructions or steps that are followed in a specific order to solve a problem. Algorithms are used in many fields, including computer science, mathematics, and logistics. An algorithm is made of well-defined steps, including mathematical operations, conditional statements, and loops executed in a specific order.

Here is a list of the most important type of algorithms to begin with:

1. Brute Force algorithm
2. Greedy algorithm
3. Recursive algorithm
4. Backtracking algorithm
5. Divide & Conquer algorithm
6. Dynamic programming algorithm
7. Randomized algorithm

Greedy Algorithm:

In this algorithm, a decision is made that is good at that point without considering the future. This means that some local best is chosen and considered as the global optimal.

There are two properties in this algorithm:

- Greedily choosing the best option
- Optimal substructure property: If an optimal solution can be found by retrieving the optimal solution to its subproblems.

Greedy Algorithm does not always work but when it does, it works like a charm! This algorithm is easy to device and most of the time the simplest one. But making locally best decisions does not always work as it sounds. So, it is replaced by a reliable solution called the Dynamic Programming approach.

Applications:

- Sorting: Selection Sort, Topological sort
- Prim's & Kruskal's algorithms
- Coin Change problem
- Fractional Knapsack Problem
- Job Scheduling algorithm

For better understanding let's go through the most common problem i.e. Job scheduling problem: Let us consider a situation where we are given the starting and end times of various events in an auditorium. Now your job is to maximize the number of events that can be organized in the auditorium where no two events overlap (starting time or ending time of one event does not fall in between the starting and endpoint of another event).

We consider six such events:

	A	B	C	D	E	F
Starting Time	1	2	5	4	3	6
Ending Time	6	3	9	6	5	10

Now a brute force solution would make us think that if we sort the events by their starting times & starting with the first event while excluding all events which overlap the previous will certainly give a solution but it won't maximize the number of events. Let us see, after sorting by starting time-

	A	B	E	D	C	F
Starting Time	1	2	3	4	5	6
Ending Time	6	3	5	6	9	10

So, the events that can be organized are- A, F. So, our brute force approach will have multiple such cases and fail if we don’t select the optimal initial event. Now let’s see what our greedy algorithm suggests. According to the greedy algorithm, we sort the events by their ending times, i.e. we select events that end first. Our new event table will become:

	B	E	A	D	C	F
Starting Time	2	3	1	4	5	6
Ending Time	3	5	6	6	9	10

So, we choose – B, E, C which is certainly a larger number of events than previous. Hence in such cases, the Greedy Algorithm gives the best solution to this type of problem.

Answer to the Question No- 2 (a)

Searching Algorithm:

Any algorithm which solves the search problem, namely, to retrieve information stored within some data structure, or calculated in the search space of a problem domain, either with discrete or continuous values.

Searching algorithms are designed to check or retrieve an element from any data structure where that element is being stored. They search for a target (key) in the search space.

Types of Search Algorithms:

Linear Search: Linear Search is defined as a sequential search algorithm that starts at one end and goes through each element of a list until the desired element is found, otherwise, the search continues till the end of the data set.

Binary Search: Binary Search is a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(Log n).

Sorting Algorithm:

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

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

Answer to the Question No- 2 (b)

Huffman Coding:

Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters.

The variable-length codes assigned to input characters are Prefix Codes, means the codes (bit sequences) are assigned in such a way that the code assigned to one character is not the prefix of code assigned to any other character. This is how Huffman Coding makes sure that there is no ambiguity when decoding the generated bitstream.

Let us understand prefix codes with a counter example.

Let there be four characters a, b, c and d, and their corresponding variable length codes be 00, 01, 0 and 1. This coding leads to ambiguity because code assigned to c is the prefix of codes assigned to a and b. If the compressed bit stream is 0001, the de-compressed output may be “cccd” or “ccb” or “acd” or “ab”.

There are mainly two major parts in Huffman Coding

- Build a Huffman Tree from input characters.
- Traverse the Huffman Tree and assign codes to characters.

Algorithm:

The method which is used to construct optimal prefix code is called Huffman coding.

This algorithm builds a tree in bottom up manner. We can denote this tree by $\underline{T}$

Let, |c| be number of leaves

|c| -1 are number of operations required to merge the nodes. Q be the priority queue which can be used while constructing binary heap.

Answer to the Question No- 2 (c)

Mathematical Algorithms:

An algorithm in math is a procedure, a description of a set of steps that can be used to solve a mathematical computation.

For example, a step-by-step procedure used in long divisions is a common example of a mathematical algorithm.

Standard Algorithm for Addition- There are four simple steps for the standard algorithm for addition:

Step 1: Line up the numbers vertically by matching the place values.

Step 2: Add together the numbers that share the same place value, starting with the one's column.

Step 3: Write the sum below each column.

Step 4: If the sum of a column is greater than 9, carry over the tens digit to the next column.

Standard Algorithm for Subtraction- There are four simple steps for the standard algorithm for addition:

Step 1: Line up the numbers vertically by matching the place values.

Step 2: Subtract the numbers that share the same place value, starting with the one's column.

Step 3: Write the difference below each column.

Step 4: If the number on the top of a column is less than the number at the bottom, then regroup before subtracting.

Graph Algorithm:

Graph algorithms are a set of instructions that traverse (visits nodes of a) graph. Some algorithms are used to find a specific node or the path between two given nodes.

Graphs are very useful data structures that can be to model various problems. These algorithms have direct applications on Social Networking sites, State Machine modeling, and many more.

Some of the most common graph algorithms are:

Breadth First Search (BFS)- Breadth First Search is one of the most simple graph algorithms. It traverses the graph by first checking the current node and then expanding it by adding its successors to the next level. The process is repeated for all nodes in the current level before moving to the next level. If the solution is found the search stops.

Depth First Search (DFS)- Depth First Search is one of the most simple graph algorithms. It traverses the graph by first checking the current node and then moving to one of its successors to repeat the process. If the current node has no successor to check, we move back to its predecessor and the process continues (by moving to another successor). If the solution is found the search stops.

Dijkstra Algorithm- Dijkstra's Algorithm is a graph algorithm presented by E.W. Dijkstra. It finds the single source shortest path in a graph with non-negative edges.

Floyd-Warshall Algorithm- Floyd-Warshall algorithm is a great algorithm for finding the shortest distance between all vertices in a graph. It has a very concise algorithm and O(V^3) time complexity (where V is a number of vertices). It can be used with negative weights, although negative weight cycles must not be present in the graph.

Evaluation: Space Complexity: O(V^2), Worse Case Time Complexity: O(V^3)

>> END <<