Victoria University of Bangladesh
Name: Md. Ziaul Hoque "Sohel"
Student ID: 2221220031
Course Title: Algorithm
Course Code: CSI 227
Batch: 22nd (Evening)
Semester: Spring-2023

Ans to the Que No 1(A)

Algorithm:

we can learn analysis techniques that allow us to compare and contrast solutions based solely on their own characteristics, not the characteristics of the program or computer used to implement them.

This article is for those who have just started learning algorithms and wondered how impactful it will be to boost their career/programming skills. It is also for those who wonder why big companies like Google, Facebook, and Amazon hire programmers who are exceptionally good at optimizing Algorithms

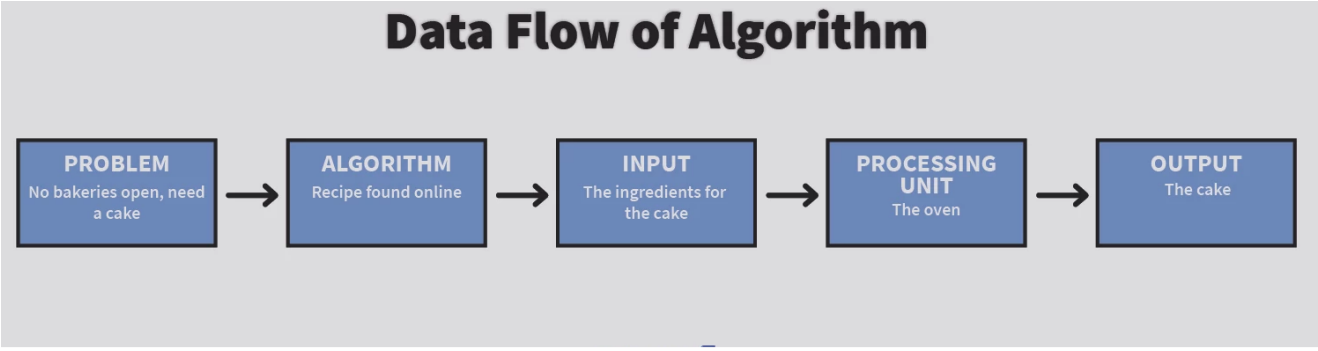
Application of Algorithm:

It is commonly used in web-based maps and games to find the shortest path at the highest possible efficiency. Algorithms is used in many artificial intelligence applications, such as search engines. It is used in other algorithms such as the Bellman-Ford algorithm to solve the shortest path problem. The algorithm is used in network routing protocols, such as RIP, OSPF, and BGP, to calculate the best route between two nodes. Arranging a particular type of data in a sequential arrangement: Storing contacts on our phone, Storing speech signals in speech processing, etc. Implementing hash tables, heaps, segment trees, etc.

Ans to the Que No 1(B)

Dataflow of Algorithms:

Let us refer to our cake baking scenario once again-
The closed bakeries due to Covid restrictions was our problem. Our recipe was the algorithm. The ingredients were our input. The oven was the processing unit and finally, our cake was the output!



Well, with this scenario, we just saw how the dataflow of an algorithm works. Now we are going to break them down and take a closer look-

- **Problem-** The problem can be any real world or a programmable problem. The problem statement usually gives the programmer an idea of the issue at hand, the available resources and the motivation to come with a plan to solve it
- **Algorithm-** After analyzing the problem, the programmer designs the step-by-step procedure to solve the problem efficiently. This procedure is the algorithm
- **Input-** The algorithm is designed and the relevant inputs are supplied
- **Processing Unit-** The processing unit receives these inputs and processes them as per the designed algorithm
- **Output-** Finally, after the processing is complete, we receive the favorable output of our problem statement.

Difference between Algorithm and Pseudocode:

The following table highlights the key differences between algorithm and pseudocode –

Algorithm	Pseudocode
It is defined as a sequence of well-defined steps. These steps provide a solution/ a way to solve a problem in hand.	It can be understood as one of the methods that helps in the representation of an algorithm.
It is a systematic, and a logical approach, where the procedure is defined step-wise	It is a simpler version of coding in a programming language.
Algorithms can be represented using natural language, flowchart and so on.	It is written in plain English, and uses short phrases to write the functionalities that s specific line of code would do.
This solution would be translated to machine code, which is then executed by the system to give the relevant output.	There is no specific syntax which is actually present in other programming languages. This means it can't be executed on a computer.
Many simple operations are combined to help form a more complicated operation, which is performed with ease by the computer	There are many formats that could be used to write pseudo-codes.
It gives the solution to a specific problem.	Most of these formats take the structure from languages such as C, LIST, FORTRAN, and so on.
It can be understood as the pseudocode for a program.	Pseudocode is not actually a programming language.
Plain text is used.	Control structures such as 'while', 'if-thenelse', 'repeat-until', and so on can be used.
It is easy to debug.	It is relatively difficult to debug.
Its construction is tough.	Its construction is easy.
There are no rules to follow while constructing it.	It has certain rules to follow while constructing it.

Ans to the Que No 1(C)

Types Of Algorithms:

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 considers it 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 Dynamic Programming approach.

Ans to the Que No 2(A)

Searching Algorithm:

Searching is the process of finding a particular item in a collection of items. A search typically answers whether the item is present in the collection or not. Searching requires a key field such as name, ID, code which is related to the target item. When the key field of a target item is found, a pointer to the target item is returned. The pointer may be an address, an index into a vector or array, or some other indication of where to find the target. If a matching key field isn't found, the user is informed.
The most common searching algorithms are:

- Linear search
- Binary search
- Interpolation search
- Hash table

Sorting Algorithm:

Sorting is the process of placing elements from a collection in some kind of order. For example, a list of words could be sorted alphabetically or by length. Efficient sorting is important to optimize the use of other algorithms that require sorted lists to work correctly.
Importance of sorting
To represent data in more readable format.
Optimize data searching to high level.

The most common sorting algorithms are:

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

Ans to the Que No 2(B)

Functions For Algorithm:

One of the most important factors one needs to take into account when designing and implementing algorithms is the time complexity that is computed during algorithm analysis.
Time complexity corresponds to the amount of time required for an algorithm to run over the provided input in order to generate the required output. In this article, we are going through the most common functions which are useful in the context of algorithm analysis. Furthermore, some code examples are provided in order to help readers understand how these functions relate to common operations undertaken by the computer, as part of the execution of an algorithm.

- Constant Function
- Linear Function
- Log Function
- Quadratic Function
- Cubic Function
- Exponential Function

Ans to the Que No 2(C)

Mathematical Algorithm:

In mathematics and computer science, an algorithm is a finite sequence of rigorous instructions, typically used to solve a class of specific problems or to perform a computation. Algorithms are used as specifications for performing calculations and data processing.

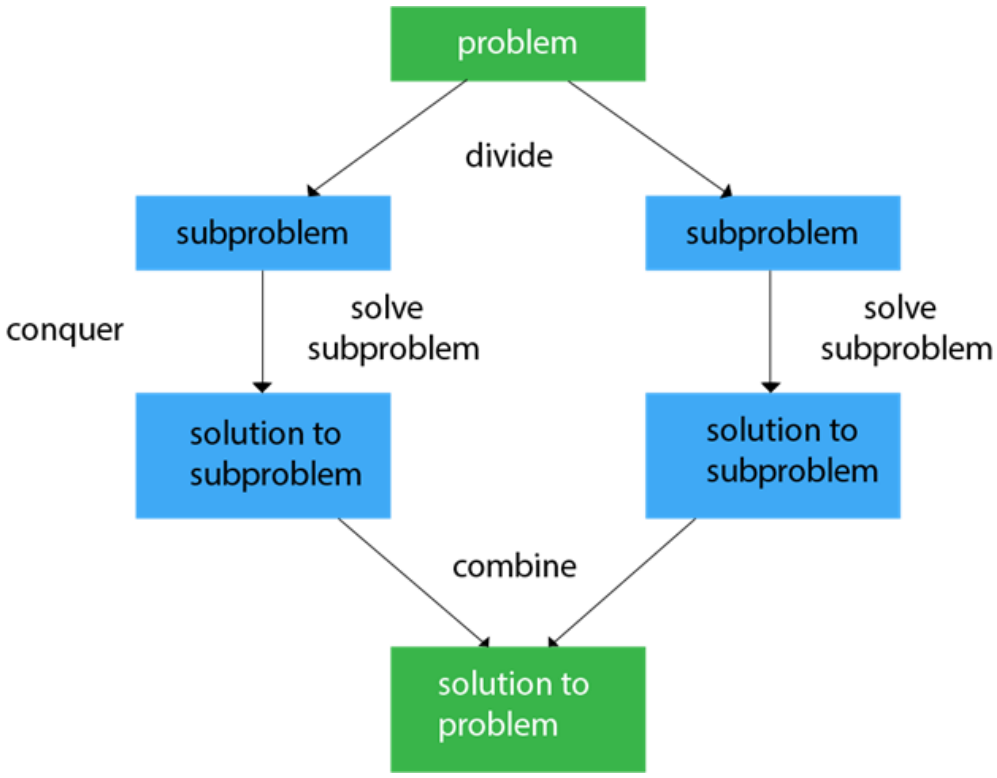
Graph Algorithm:

A graph is a unique data structure in programming that consists of finite sets of nodes or vertices and a set of edges that connect these vertices to them. At this moment, adjacent vertices can be called those vertices that are connected to the same edge with each other. In simple terms, a graph is a visual representation of vertices and edges sharing some connection or relationship. Although there are plenty of graph algorithms that you might have been familiar with, only some of them are put to use. The reason for this is simple as the standard graph algorithms are designed in such a way to solve millions of problems with just a few lines of logically coded technique. To some extent, one perfect algorithm is solely optimized to achieve such efficient results.

Divide and Conquer Algorithm:

Divide and conquer is an algorithmic pattern. In algorithmic methods, the design is to take a dispute on a huge input, break the input into minor pieces, decide the problem on each of the small pieces, and then merge the piecewise solutions into a global solution. This mechanism of solving the problem is called the Divide & Conquer Strategy.
Divide and Conquer algorithm consists of a dispute using the following three steps.

1. **Divide** the original problem into a set of subproblems.
2. **Conquer:** Solve every subproblem individually, recursively.
3. **Combine:** Put together the solutions of the subproblems to get the solution to the whole problem.



----------END----------