COMPUTER SCIENCE 10TH – DETAILED QUESTION ANSWERS

PROBLEM SOLVING AND ALGORITHM DESIGNING

Chapter # 01

Q.1: What is problem solving?

Ans. PROBLEM SOLVING:

Problem solving is a procedure to figure out the solution of complex problems. Problem solving is the main process in computer programming, where programmers first understand the problem, plan the solution and then understand how to translate algorithm into something a computer can do, and finally how to write the specific syntax or code required by a computer to get the job done.

Q.2: Define Problem?

Ans. PROBLEM:

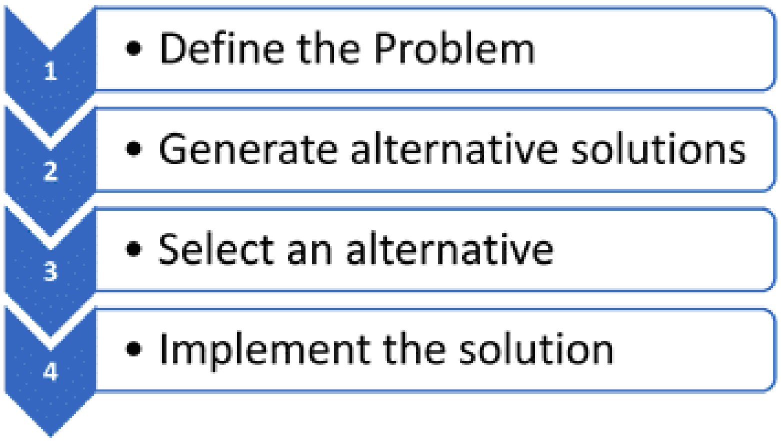
Problem sometimes referred as an issue, is any situation that occurs and is unexpected or prevents something from occurring. In programming a problem is considered as a matter which is difficult to solve, it is a complex task or routine.

Q.3: What are the strategies of problem solving?

Ans. PROBLEM SOLVING PROCESS:

Finding a suitable solution for issues can be accomplished by the following the basic four step problem-solving process given below.

1. Define the problem.
2. Generate alternative solutions.
3. Evaluate and select an alternative.
4. Implement and follow up on the solution.



1. DEFINE THE PROBLEM

   Specifically, the task of defining the problem consists of identifying what you know (input-given data), and what you want to obtain (output-the result). During this step, it is more important to understand described problem.

2. GENERATE ALTERNATIVE SOLUTIONS

   Many alternative solutions to the problem should be generated to achieve the desired results. Generating multiple alternatives can enhance the value of best solution. So, it is good to generate a list of all possible solutions and evaluate to decide which one is the best for that particular problem.

3. EVALUATE AND SELECT AN ALTERNATIVE

   Good problem-solving approach uses a series of considerations when selecting the best alternative. In this step, evaluate all possible solutions related to a target standard and select the best alternative. A particular alternative will solve the problem without causing other unanticipated problems.

4. IMPLEMENT AND FOLLOW UP ON THE SOLUTION

   The implementation of best solution also includes planning on what happens when something goes wrong with the selected solution. It is also important to follow up and track the results of solution at every stage of implementation.

Q.4: What is algorithm?

Ans. ALGORITHM

An algorithm is a procedure or formula for solving a problem. It can be defined as the set of well-defined step by step instructions to solve a program. In computer science an algorithm usually means a small procedure that solves a specific problem designs; algorithms are widely used throughout all areas of IT. There are two common methods to represent algorithm designs; pseudocode and flowcharts.

Q.5: Explain the role of algorithm in problem solving. OR Describe the advantages of algorithm.

Ans. ROLE OF ALGORITHM IN PROBLEM SOLVING

Algorithms give us the most ideal option of accomplishing a task in problem solving. Algorithms are used to find the best possible way of solving a problem. In doing so they improve the efficiency of a program. With the best algorithm, a computer program will be able to produce very accurate results. An algorithm can be used to improve the speed at which a program executes.

Q.6: What are the qualities of a good algorithm?

Ans. QUALITIES OF A GOOD ALGORITHM

1. Inputs and outputs should be defined exactly.
2. Every step in an algorithm must be clear, precise and unambiguous.
3. It should be effective among many different ways to solve a problem.
4. It shouldn't have computer code. Instead, the algorithm must be written in such a way that, it can be used in similar programming languages.

Q.7: Write some examples of algorithm.

Ans. ALGORITHM EXAMPLES

ALGORITHM 1: MAKING A CUP OF TEA

```
Step 1: Start.
Step 2: Put the teabag in a cup.
Step 3: Fill the kettle with water.
Step 4: Boil the water in the kettle.
Step 5: Pour some of the boiled water into the cup.
Step 6: Add milk to the cup.
Step 7: Add sugar to the cup.
Step 8: Stir the tea.
Step 9: Drink the tea.
Step 10: Stop
```

ALGORITHM 2: SUBTRACTION OF TWO NUMBERS

```
Step 1: Start
Step 2: Declare variables n1, n2 & sub
Step 3: Read values n1 & n2
Step 4: Subtract n1 & n2 and assign the result to sub
        sub = n1 - n2
Step 5: Display sub
Step 6: Stop
```

ALGORITHM 3: ADDITION OF TWO NUMBERS

```
Step 1: Start
Step 2: Declare variables n1, n2 & sum
Step 3: Read values n1 & n2
Step 4: Add n1 & n2 and assign the result to sum
        sum = n1 + n2
Step 5: Display sum
Step 6: Stop
```

ALGORITHM 4: MULTIPLICATION OF TWO NUMBERS

```
Step 1: Start
Step 2: Declare variables n1, n2 & mul
Step 3: Read values n1 & n2
Step 4: Multiply n1 & n2 and assign the result to mul
        mul = n1 * n2
Step 5: Display mul
Step 6: Stop
```

ALGORITHM 5: AREA OF CIRCLE

```
Step 1: Start
Step 2: Declare variable r and area
Step 3: Read value of r
Step 4: Apply formula {Area = (3.142) * r * r}
        area = (3.142) * r * r
Step 5: Display area
Step 6: Stop
```

Q.8: What is flowchart?

FLOWCHART

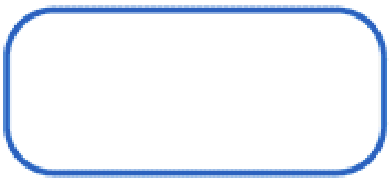
A flowchart is a pictorial representation of an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. It is used to show the sequence of steps and logic of solution to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

Q.9: Define different symbols used in flowchart.

Ans. FLOWCHART SYMBOLS

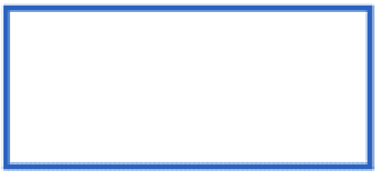
1. TERMINAL OR START / STOP

The terminal symbol is used to indicate the beginning (START) and ending (STOP) in the program logic flow.



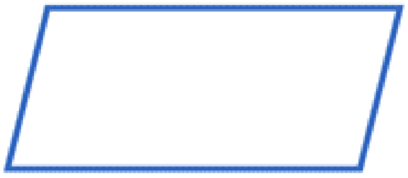
2. Process

Process symbol is used to illustrate a process, action or an operation. These are represented by rectangles.



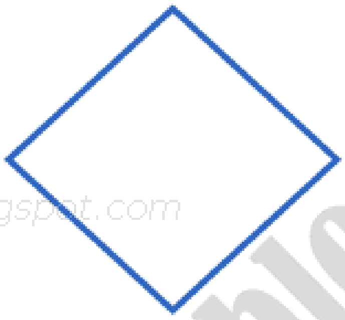
3. Input / Output

The input / output symbol is used to denote any type of input data or output information.



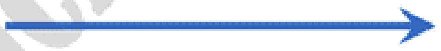
4. Decision

The decision symbol is used in a flowchart to indicate a point at which a decision has to be made and a branch to one of two or more alternative points are possible.



5. Arrows or Flowlines

Flowlines with arrowheads are used to indicate the flow of operation. It shows the exact sequence in which the instructions are to be executed.



Q.10: Write down the importance of flowchart in problem solving.

Ans. IMPORTANCE OF FLOWCHART IN SOLVING A PROBLEM

1. Communication

   Flowcharts are better way of communicating the logic of a system.

2. Effective Analysis

   With the help of flowchart, problem can be analyzed in more effective way therefore reducing cost and wastage of time.

3. Efficient Coding

   The flowcharts act as a guide or blueprint during the systems analysis and program development phase.

4. **Proper Debugging**

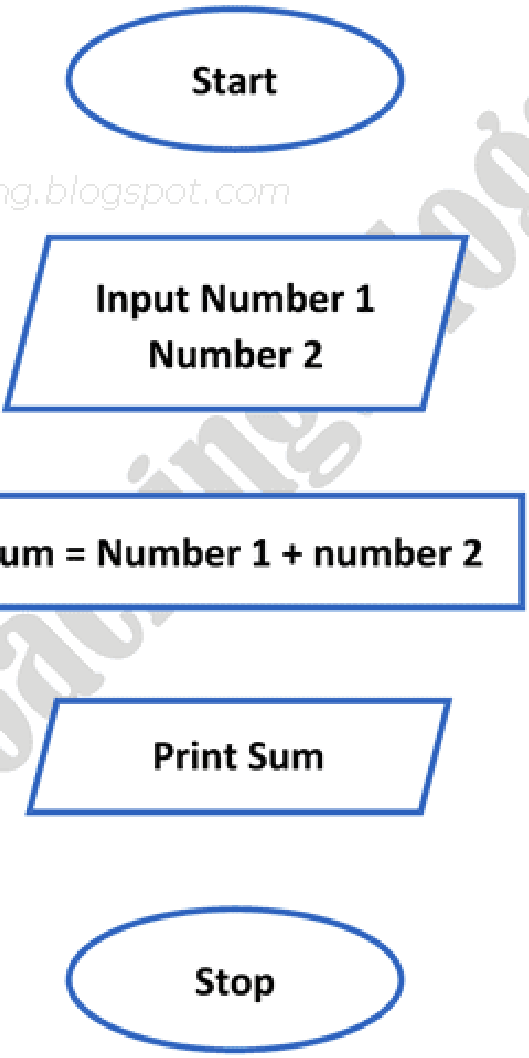
   The flowchart helps in debugging a process.

5. **Efficient Program Maintenance**

   The maintenance of operating program becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part.

Q.11: Write down the example of flowchart.

Ans. FLOWCHART EXAMPLE



Q.12: What are the differences between algorithm and flowchart?

Ans. DIFFERENCES BETWEEN ALGORITHM & FLOWCHART

ALGORITHM	FLOWCHART
It is step by step solution of program	It is the diagrammatical representation which shows the flow of data.
Text in common language is used in algorithm.	Symbols and shapes are used in flowchart.
It is difficult to write and understand.	It is easy to construct and understand
Algorithm doesn't have any specific rules.	Flowchart have specific rules for its construction.

Q.13: Define data structures.

Ans. Data Structures:

Data structure is a particular means of organizing and storing data in computers in such a way that we can perform operations on the stored data more efficiently. Data structures have a wide and diverse scope of usage across the fields of Computer Science and Software engineering. Data structure may be linear or non-linear.

Q.14: Describe linear data structure.

Ans: LINEAR DATA STRUCTURE

Linear data structure arranges the data in a linear fashion. The data elements are arranged sequentially such that the element is directly linked to its previous and the next elements. This data structure supports single-level storage of data. And hence, transfer of the data is achieved through a single run only.

Examples of linear data structures are Stack, Queue, Array, etc.

Q.15: Define stack and its operations.

Ans. STACK

Stack is a linear data structure that stores information in such a way that the last item stored is the first item retrieved. It is based on the principle of LIFO (Last-in-first-out). The stack in digital computers is a group of memory locations with a register that holds the address of top of element. Stacks usually performs two operations.

Push: It is used to insert an element on top of stack.

Pop: It is used to delete an element from top of stack.

Q.16: Define queue data structure and its operations.

Ans. QUEUE:-

Queue is the type of linear data structure where the elements to be stored follow the rule of First in First Out (FIFO). In queue, the element that was added first is removed first) In this data structure, both ends are used for the insertion and the removal of data. The two main operations of queue are enqueue and dequeue.

Enqueue: It refers to the process where inserting an element is allowed to the collection of data.

Dequeue: It refers to the process where removal of elements is allowed, which is the first element of queue.

Q.17: Define array data structure and its operations.

Ans. ARRAY:

The array is the type of linear data structure that stores homogeneous (same type) elements at memory locations which are consecutive. The same types of objects are stored sequentially in an array. The main idea of an array is that multiple data of the same type can be stored together. Before storing the data in an array, the size of the array has to be defined. The location of elements stored in an array has a numerical value called index to identify the element. Some operations of array are:

Traverse: Go through the elements and print them.

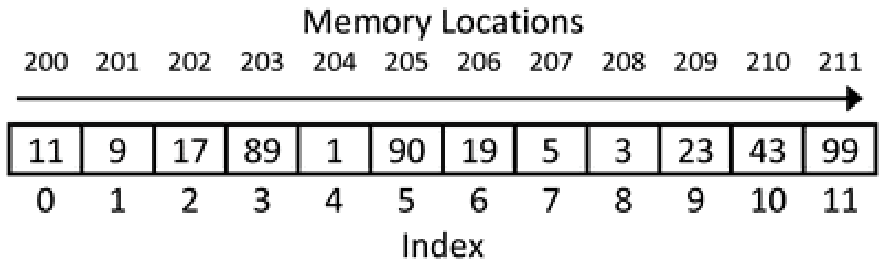
Search: Search for an element in the array by its value or its index.

Update: Update the value of an existing element at a given index.

Insert: Insert an element in array.

Deletion: Delete an element from array.

Sorting: Rearrange a given array or list of elements.



Q.18: Describe nonlinear data structure.

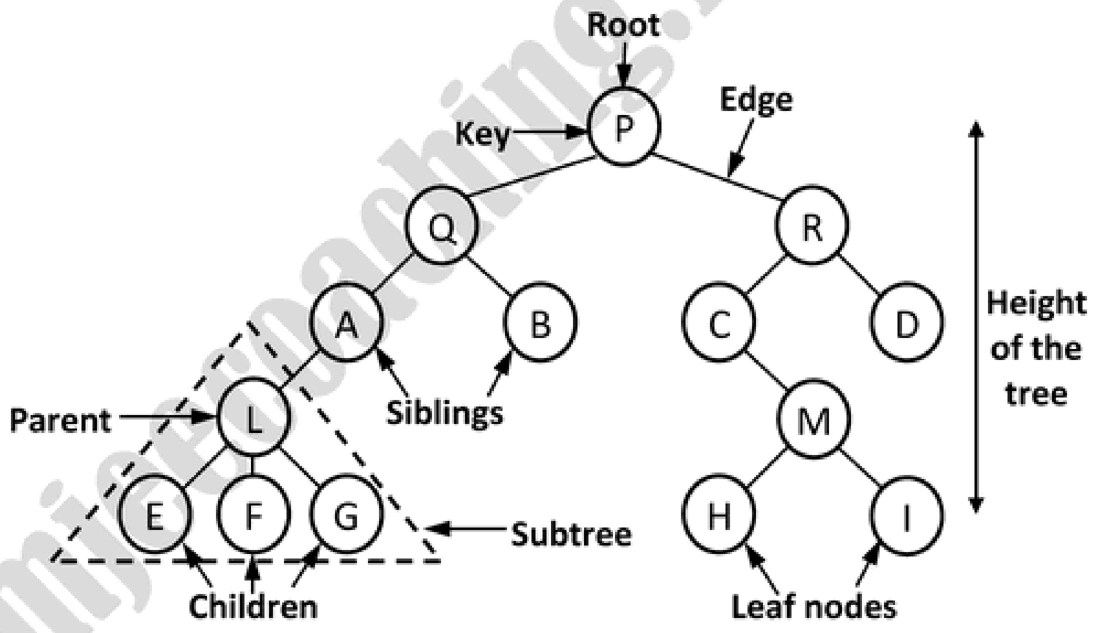
Ans. NON-LINEAR DATA STRUCTURES:

Non-linear data structures are those data structures in which data items are not arranged in a sequence. In this structure, each element can have multiple paths to connect to other elements. Non-linear data structures allow multi-level storage. Non-linear data structures are not easy to implement but are more efficient in utilizing computer memory. Examples of non-linear data structures are Tree, Graphs, etc.

Q.19: Define tree data structure.

Ans. TREE:

A tree can be defined as a finite set of data items (nodes) in which data items are arranged in branches and sub-branches. It consists of various linked nodes and has a hierarchical tree structure that forms a parent-child relationship. In tree structure, the first node is called root node. The node connected to root node are called parent node and another node connected to parent node is called child node. Various types of trees have been developed throughout the past decades, one of them are binary tree or binary search tree.



Q.20: Define graph data structure.

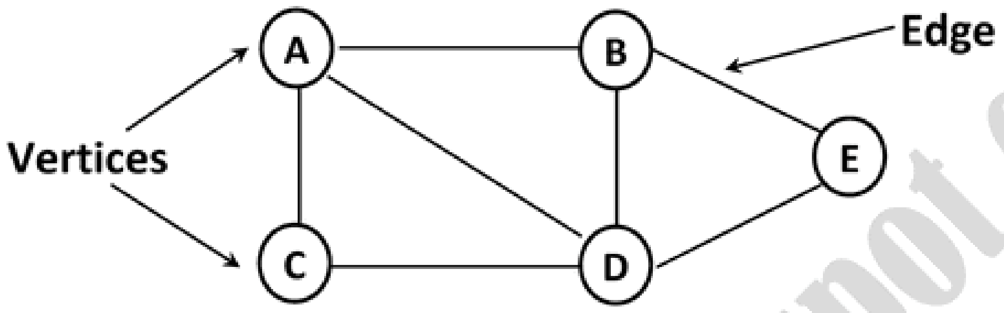
Ans. GRAPH

Graphs are used to solve many real-life problems. A graph is a non-linear data structure that has a finite number of vertices and edges, and these edges are used to connect the vertices. The vertices are used to store the data elements, while the edges represent the relationship between the vertices. The order of a graph is the number of vertices in the graph. The size of a graph is the number of edges in the graph. Two nodes are said to be adjacent if they are connected to each other by the same edge.

There are two types of graphs:

(a) Undirected graph

(b) Directed graph

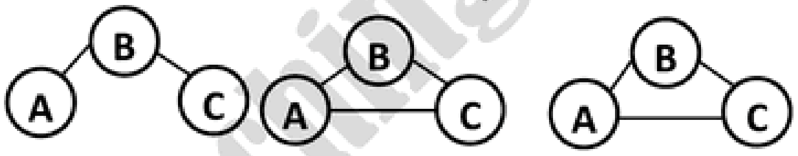


Q.21: Define undirected graph.

Ans. UNDIRECTED GRAPH

An undirected graph is a graph in which all its edges have no direction. It can go in both ways between the two vertices.

Undirected Graphs



Q.22: Define directed graph.

Ans. DIRECTED GRAPH

A directed graph is a graph in which all its edges have a direction indicating what is the start vertex and what is the end vertex.

Directed Graphs