

DATA STRUCTURE

**All Notes Are Verified by NPTEL and Published
By**

WWW.ASKTOHOW.COM

INTRODUCTION

It is important for every Computer Science student to understand the concept of *Information* and how it is organized or how it can be utilized.

What is Information?

If we arrange some data in an appropriate sequence, then it forms a Structure and gives us a meaning. This meaning is called *Information*. The basic unit of Information in Computer Science is a bit, Binary Digit.

So, we found two things in Information: One is **Data** and the other is **Structure**.

What is Data Structure?

1. A **data structure** is a systematic way of organizing and accessing data.
2. A **data structure** tries to structure data!
 - Usually more than one piece of data
 - Should define legal operations on the data
 - The data might be grouped together (e.g. in an linked list)
3. When we define a data structure we are in fact creating a new data type of our own.
 - i.e. using predefined types or previously user defined types.
 - Such new types are then used to reference variables type within a program

Why Data Structures?

1. Data structures study how data are stored in a computer so that operations can be implemented efficiently
2. Data structures are especially important when you have a large amount of information
3. Conceptual and concrete ways to organize data for efficient storage and manipulation.

Methods of Interpreting bit Setting

1. Binary Number System
 - Non Negative
 - Negative
 - Ones Complement Notation
 - Twos Complement Notation
2. Binary Coded Decimal
3. Real Number
4. Character String

Non-Negative Binary System

In this System each bit position represents a power of 2. The right most bit position represent 2^0 which equals 1. The next position to the left represents $2^1 = 2$ and so on. An Integer is represented as a sum of powers of 2. A string of all 0s represents the number 0. If a 1 appears in a particular bit position, the power of 2 represented by that bit position is included in the Sum. But if a 0 appears, the power of 2 is not included in the Sum. For example 10011, the sum is $2^0 + 2^1 + 2^4 = 19$

Ones Complement Notation

Negative binary number is represented by ones Complement Notation. In this notation we represent a negative number by changing each bit in its absolute value to the opposite bit setting. For example, since 001001100 represent 38, 11011001 is used to represent -38. The left most number is reserved for the sign of the number. A bit String Starting with a 0 represents a positive number, where a bit string starting with a 1 represents a negative number.

Twos Complement Notation

In Twos Complement Notation is also used to represent a negative number. In this notation 1 is added to the Ones Complement Notation of a negative number. For example, since 11011001 represents -38 in Ones Complement Notation 11011010 used represent -38 in Twos Complement Notation.

Binary Coded Decimal

In this System a string of bits may be used to represent integers in the *Decimal Number System* . Four bits can be used to represent a Decimal digit between 0 and 9 in the binary notation. A string of bits of arbitrary length may be divided into consecutive sets of four bits. With each set representing a decimal digit. The string then represents the number that is formed by those decimal digits in conventional decimal notation. For example, in this system the bit string 00110101 is separated into two strings of four bits each: 0011 and 0101. The first of these represents the decimal digit 3 and the second represents the decimal 5, so that the entire string represents the integer 35.

In the binary coded decimal system we use 4 bits, so this four bits represent sixteen possible states. But only 10 of those sixteen possibilities are used. That means, whose binary values are 10 or larger, are invalid in *Binary Coded Decimal System*.

Real Number

The Floating Point Notation use to represent Real Numbers. The key concept of floating-point notation is *Mantissa*, *Base* and *Exponent* . The base is usually fixed and the Mantissa and the Exponent vary to represent different Real Number. For Example, if the base is fixed at 10, the number -235.47 could be represented as $-23547 * 10^{-2}$. The Mantissa is 23547 and the exponent is -2. Other possible representations are $-.23547 * 10^3$ and $-235.47 * 10^0$.

In the floating-point notation a real number is represented by a 32-bit string. Including in 32-bit, 24-bit use for representation of Mantissa and remaining 8-bit use for representation of Exponent .Both the mantissa and the exponent are twos complement binary Integers. For example, the 24-bit twos complement binary representation of -23547 is 111111111010010000000101, and the 8-bit twos complement binary representation of -2 is 11111110. So the representation of 235.47 is 11111111101001000000010111111110. It can be used to represent extremely large or extremely small absolute values.

Character Strings

In computer science, information is not always interpreted numerically. Item such as names, address and job title must also be represented in some fashion with in computer. To enable the representation of such nonnumeric objects, still another method of interpreting bit strings is necessary. Such information is usually represented in character string form. For example, in some computers, the eight bits 11000000 is used to represent the character "A" and 11000001 for character "B" and another for each character that has a representation in a particular machine. So, the character string "AB" would be represented by the bit string 1100000011000001.

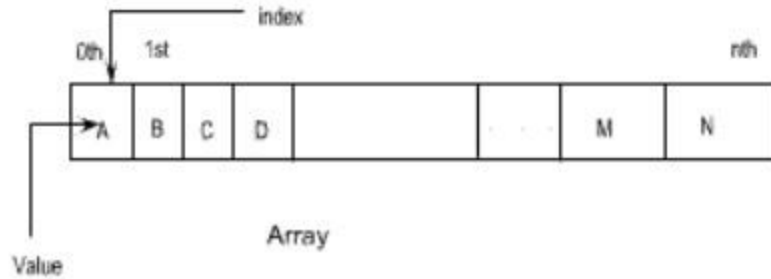
The assignment of bit string to character may be entirely arbitrary, but it must be adhered to consistently. It may be that some convenient rule is used in assigning bit string to character. The number of bits varies computer wise used to represent a character.

Some computers are use 7-bit (therefore allow up to 128 possible characters), some computers are use 8-bits (up to 256 character), and some use 10-bits (up to 1024 possible characters). The number of bits necessary to represent a character in a particular computer is called the **byte size** and a group of bits that number is called a **byte** .

Array

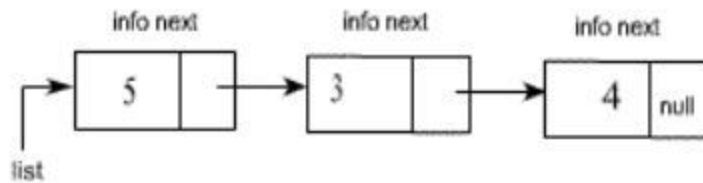
In computer programming, a group of homogeneous elements of a specific data type is known as an array , one of the simplest data structures. Arrays hold a series of data elements, usually of the same size and data type. Individual elements are accessed by their position in the array. The position is given by an index, which is also called a subscript. The index usually uses a consecutive range of integers, (as opposed to an associative array) but the index can have any ordinal set of values.

Some arrays are *multi-dimensional* , meaning they are indexed by a fixed number of integers, for example by a tuple of four integers. Generally, one- and two-dimensional arrays are the most common. Most programming languages have a built-in *array* data type.



Link List

In computer science, a **linked list** is one of the fundamental data structures used in computer programming. It consists of a sequence of nodes, each containing arbitrary data fields and one or two references ("links") pointing to the next and/or previous nodes. A linked list is a self-referential data type because it contains a link to another data of the same type. Linked lists permit insertion and removal of nodes at any point in the list in constant time, but do not allow random access.



Types of Link List

1. Linearly-linked List
 - Singly-linked list
 - Doubly-linked list
2. Circularly-linked list
 - Singly-circularly-linked list
 - Doubly-circularly-linked list
3. Sentinel nodes

Stack

A stack is a linear Structure in which item may be added or removed only at one end. There are certain frequent situations in computer science when one wants to restrict insertions and deletions so that they can take place only at the beginning or the end of the end of the list, not in the middle. Two of the Data Structures that are useful in such situations are *Stacks* and *queues*. A stack is a list of elements in which an elements may be inserted or deleted only at one end, called the *Top*. This means, in particular, the elements are removed from a stack in the reverse order of that which they are inserted in to the stack. The stack also called "*last-in first -out (LIFO)* " list.

Special terminology is used for two basic operation associated with stack :

1. "Push" is the term used to insert an element into a stack.
2. "Pop" is the term used to delete an element from a stack.

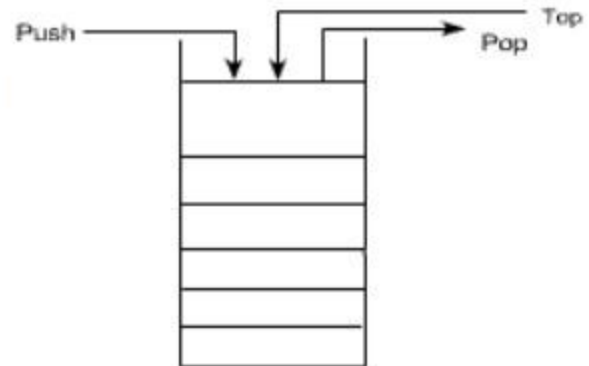
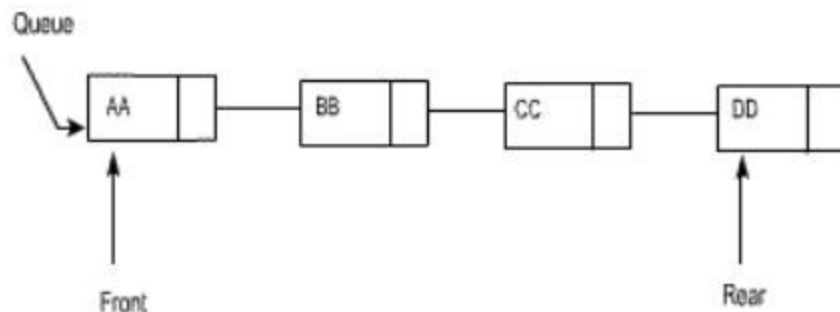


Fig: Stack

Queue

A queue is a linear list of elements in which deletions can take place only at one end, called the "*front*" and insertion can take place only at the other end, called "*rear*". The term "*front*" and "*rear*" are used in describing a linear list only when it is implemented as a queue.

Queues are also called "*first-in first-out*" (FIFO) list. Since the first element in a queue will be the first element out of the queue. In other words, the order in which elements enter in a queue is the order in which they leave. The real life example: the people waiting in a line at Railway ticket Counter form a queue, where the first person in a line is the first person to be waited on. An important example of a queue in computer science occurs in timesharing system, in which programs with the same priority form a queue while waiting to be executed.



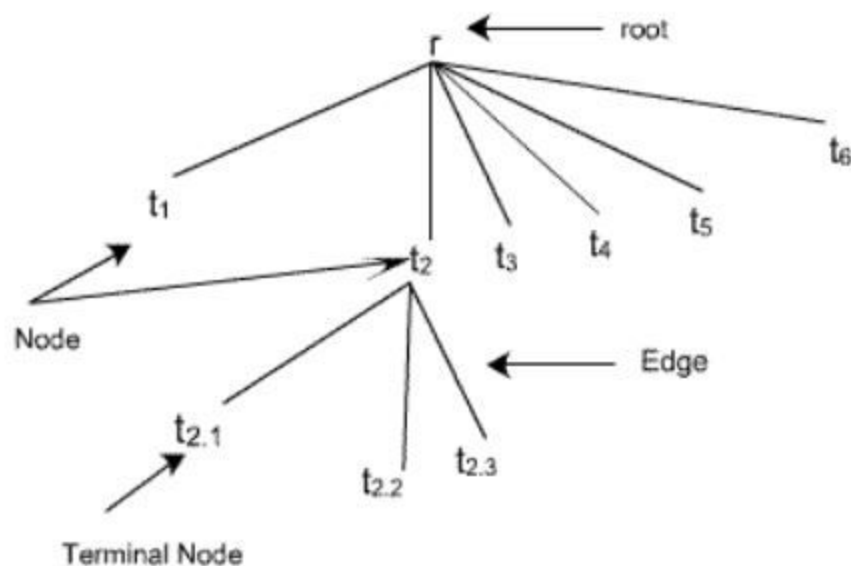
Tree

Data frequently contain a hierarchical relationship between various elements. This non-linear Data structure which reflects this relationship is called a rooted tree graph or, tree.

This structure is mainly used to represent data containing a hierarchical relationship between elements, e.g. record, family tree and table of contents.

A tree consist of a distinguished node r , called the **root** and zero or more (sub) tree t_1, t_2, \dots, t_n , each of whose roots are connected by a directed edge to r .

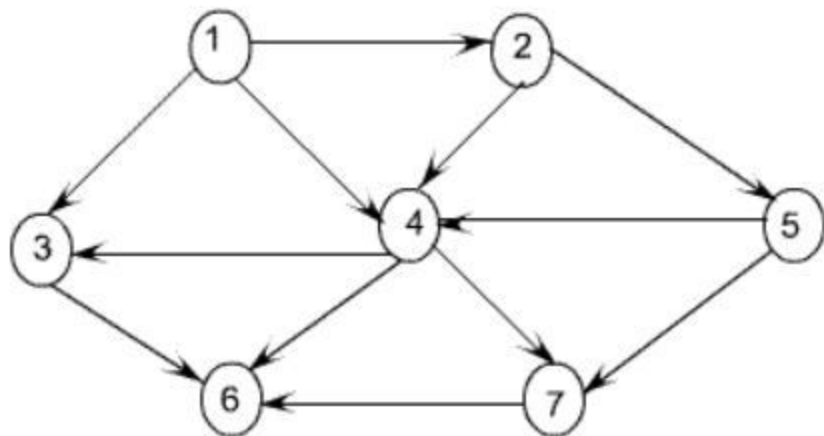
In the tree of figure, the root is A, Node t 2 has r as a parent and t 2.1, t 2.2 and t 2.3 as children. Each node may have arbitrary number of children, possibly zero. Nodes with no children are known as leaves.



Graph

A *graph* consists of a set of *nodes* (or *Vertices*) and a set of *arc* (or *edge*). Each arc in a graph is specified by a pair of nodes. A node n is *incident* to an arc x if n is one of the two nodes in the ordered pair of nodes that constitute x . The *degree* of a node is the number of arcs incident to it. The *indegree* of a node n is the number of arcs that have n as the head, and the outdegree of n is the number of arcs that have n as the tail.

The graph is the nonlinear data structure. The graph shown in the figure represents 7 vertices and 12 edges. The Vertices are $\{ 1, 2, 3, 4, 5, 6, 7 \}$ and the arcs are $\{(1,2), (1,3), (1,4), (2,4), (2,5), (3,4), (3,6), (4,5), (4,6), (4,7), (5,7), (6,7) \}$. Node (4) in figure has indegree 3, outdegree 3 and degree 6.



Abstract Data Type

1. Abstract Data Types (ADT's) are a model used to understand the design of a data structure
2. '*Abstract*' implies that we give an implementation-independent view of the data structure
3. ADTs specify the type of data stored and the operations that support the data
4. Viewing a data structure as an ADT allows a programmer to focus on an idealized model of the data and its operations

Problems :

1. What is Information in Computer Science?
2. What are methods for representing negative binary number? The following numbers convert to ones complement and twos complement notation.
 - o 00110111
 - o 01100110
 - o 01111101
 - o 10001001
3. Write a C program where following numbers are stored in a array :

2 12 17 24 5 78 35 18 16
4. Write a C program using linked list where following numbers are stored :

2 12 17 24 5 78 35 18 16

5. Consider the stack NAME in fig 1.01, which is stored alphabetically.

- Suppose Nirmal is to be inserted in to the stack. How many name must be moved to the new location?
- Suppose Sourav is to be deleted from the stack. How many names must be removed to the new location?

NAME	
1	Abhisek
2	Anupam
3	Charls
4	Debasis
5	Pranay
6	Ritwik
7	Sourav
8	Suman

fig 1.01

6. The following is a tree structure given by means of level numbers as discussed below:

01 Employee 02 Name 02 Emp. Code 02 Designation 03 Project Leader 03 Project Manager 02 Address

Draw the corresponding tree diagram.