Introduction to Data Structure

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.

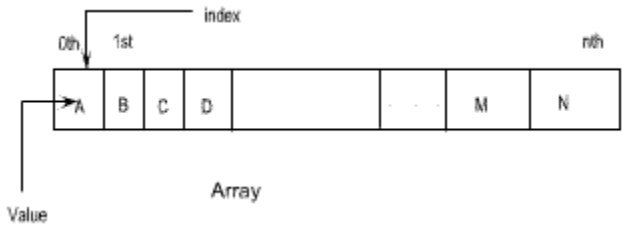
Why Data Structures?

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.

Array

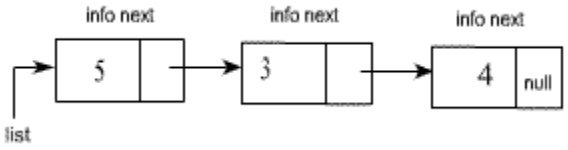
In computer programming, a group of homogeneous elements of a specific data type is known as an array, one of the simplest data structures. Arrays hold a series of data elements, usually of the same size and data type. Individual elements are accessed by their position in the array. The position is given 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 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.

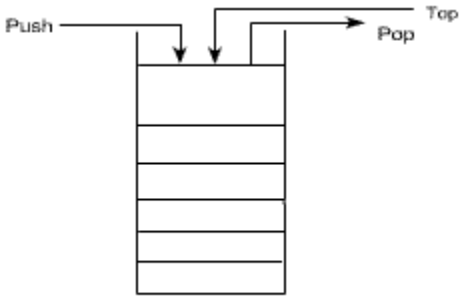


Fig: Stack

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

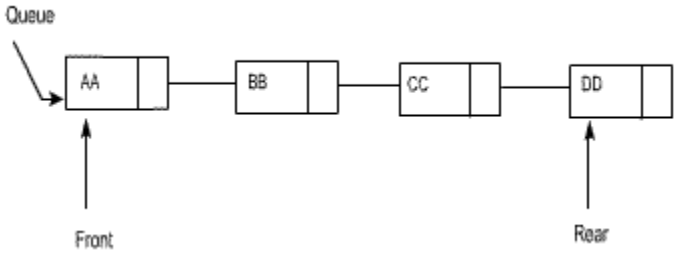
1. "Push" is the term used to insert an element into a stack.
2. "Pop" is the term used to delete an element from a stack.

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.



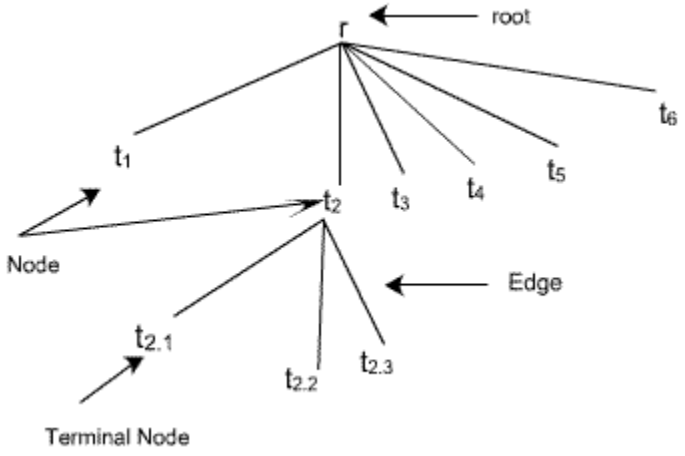
Tree

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 , ... t_n , each of whose roots are connected by a directed edge to *r* .

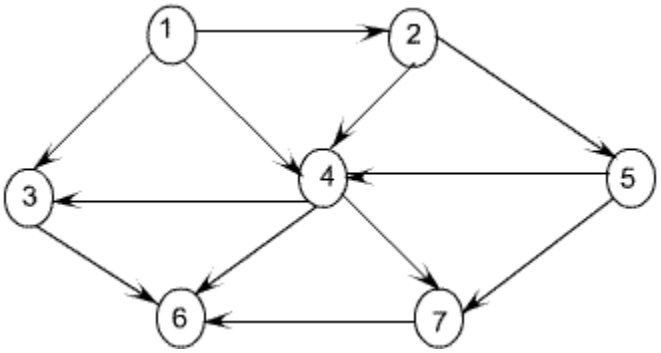
In the tree of figure, the root is A, Node t_2 has *r* as a parent and $t_{2.1}$, $t_{2.2}$ and $t_{2.3}$ as children. Each node may have arbitrary number of children, possibly zero. Nodes with no children are known as leaves.



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.



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.
    - 00110111
    - 01100110
    - 01111101
    - 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.