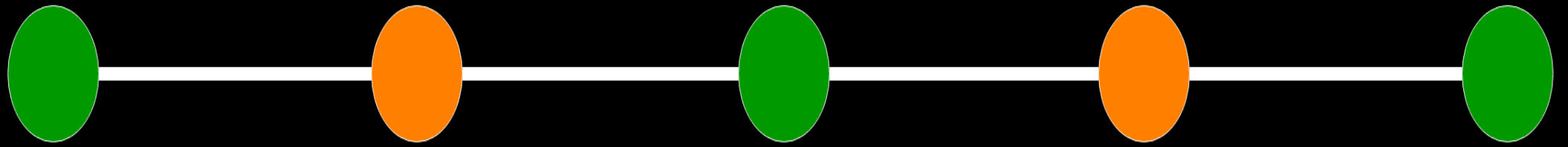
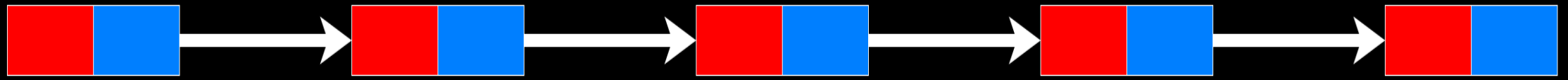


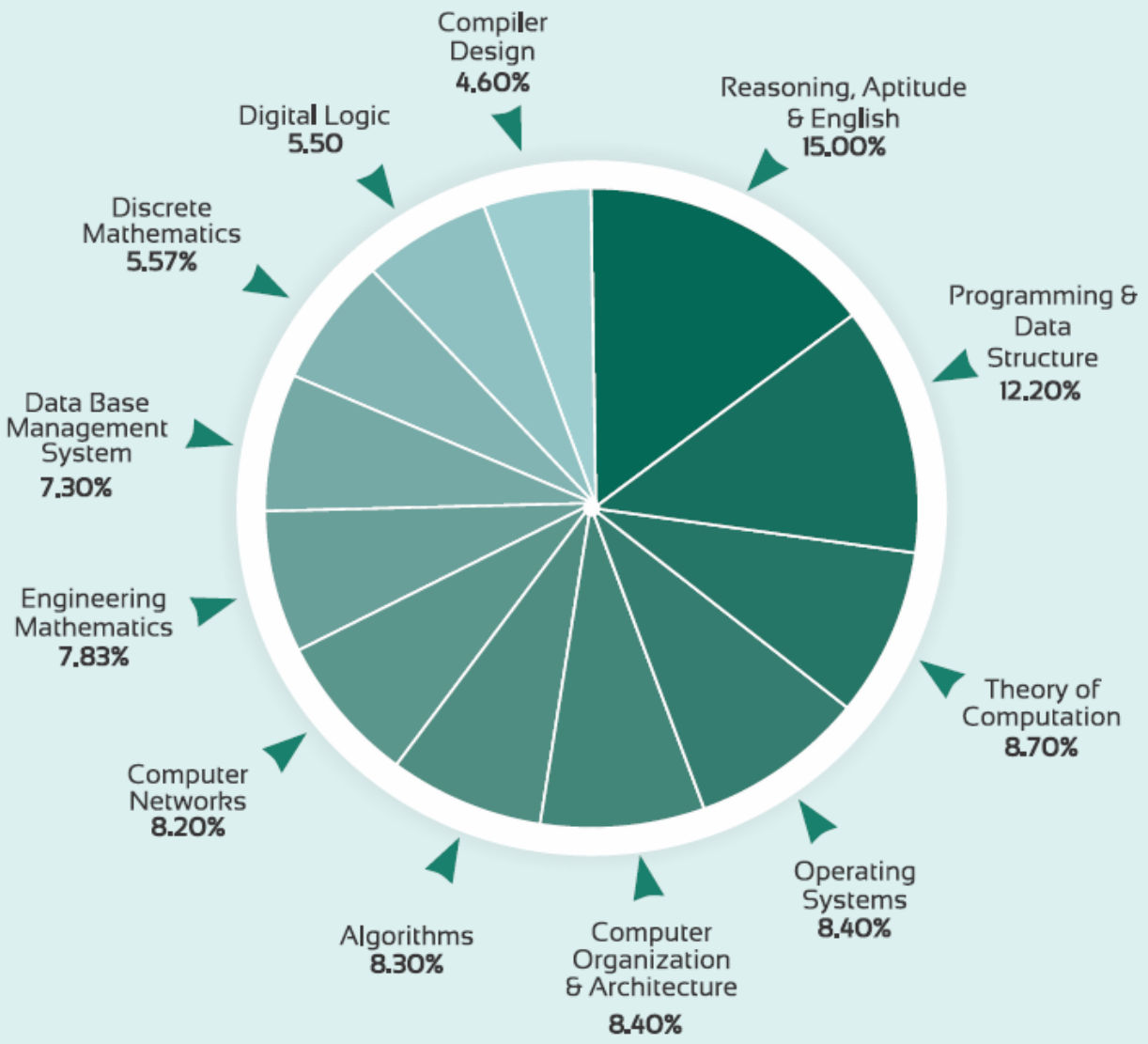
DATA STRUCTURES



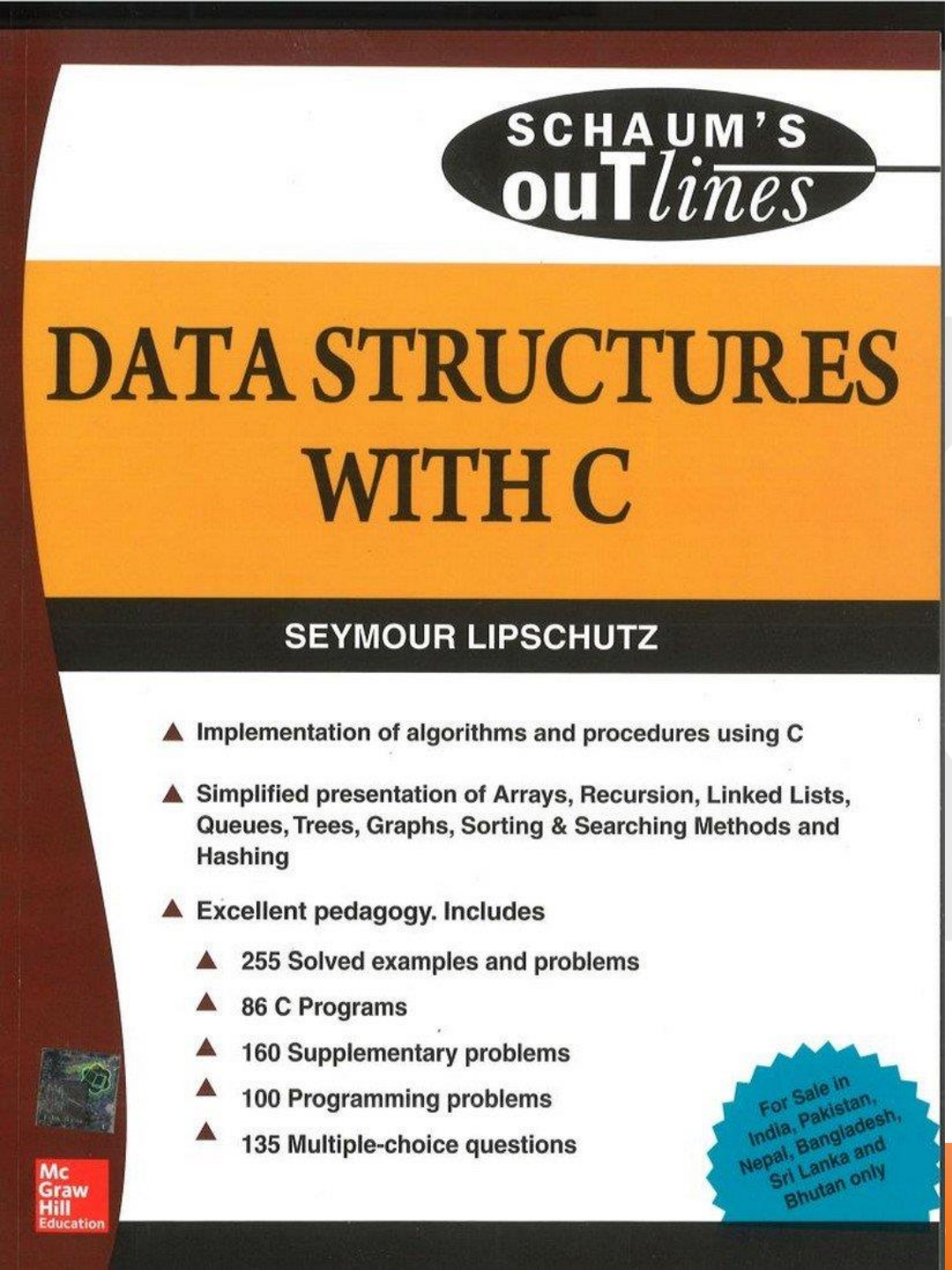
Data Structure

- Core subjects for CS/IT Students
- In GATE 7-8 Marks out of 100 Marks, and 5-6 questions on an average
- In NET 14-15 Marks out of 200 marks and 8-10 questions
- Both parts are important Theory and Numerical
- Needs a little time, good scoring
- Very important in Industry

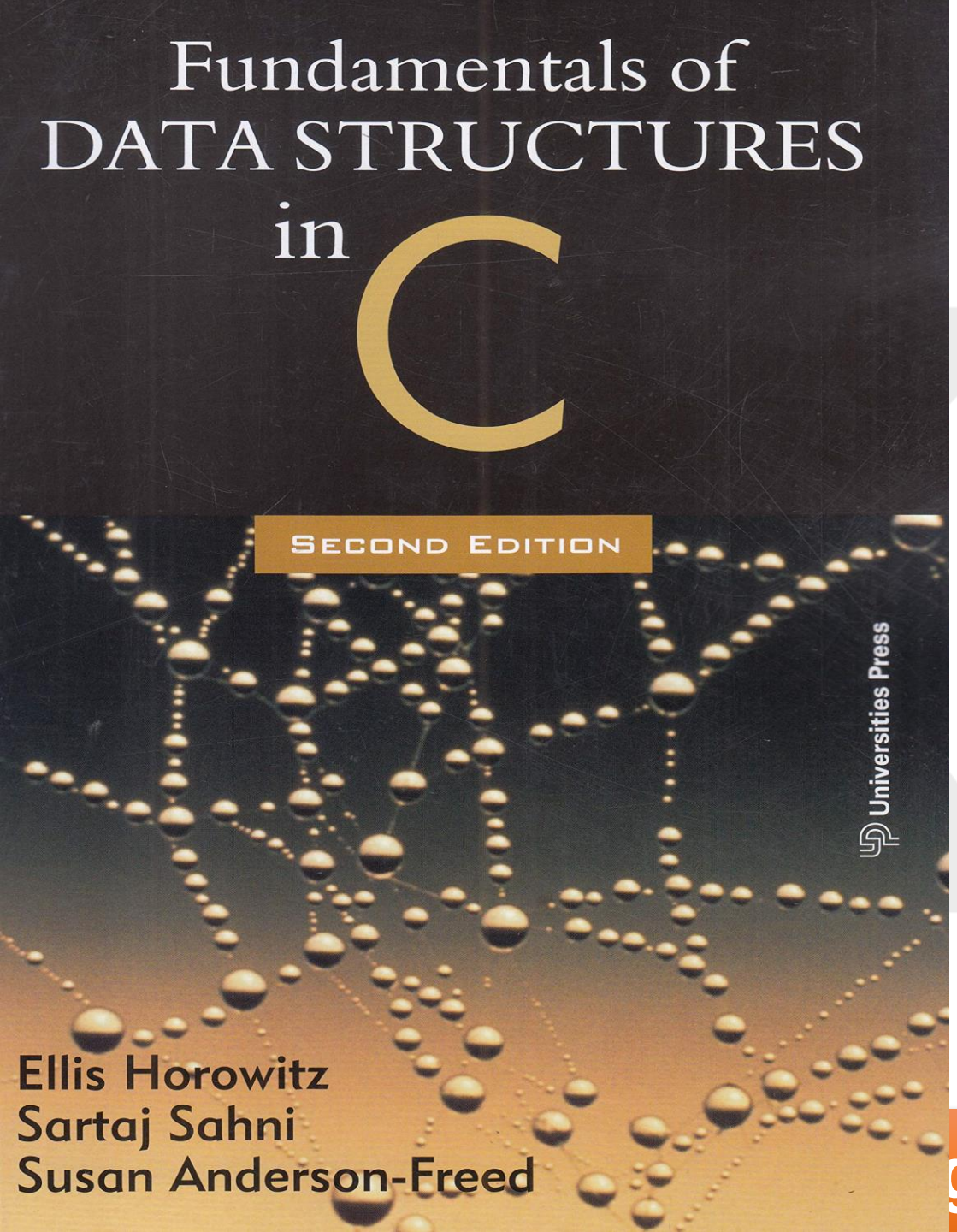
SUBJECT-WISE WEIGHTAGE ANALYSIS OF GATE SYLLABUS



Subject	Average % (last 5 yrs)
Reasoning, Aptitude & English	15.00%
Programming & Data Structure	12.20%
Computer Networks	8.20%
Theory of Computation	8.70%
Operating Systems	8.40%
Computer Organization & Architecture	8.40%
Algorithms	8.30%
Engineering Mathematics	7.83%
Data Base Management System	7.30%
Digital Logic	5.50%
Discrete Mathematics	5.57%
Compiler Design	4.60%
Total	**100%**



- Book Name
  - Data Structures with C (Schaum's Outline Series)
- Writers – Seymour Lipschutz
- Publisher – McGraw Hill



- Book Name – Fundamentals of Data Structures in C

- Writers -
  - Ellis Horowits
  - Sartaj Sahni
  - Susan Anderson-Freed

- Publisher – University Press

- Edition – 2th

Syllabus

- Arrays
- Stacks
- Queues
- Linked lists
- Trees
- Binary search trees
- Binary heaps
- Graphs

Break

https://www.amazon.in/Data-Structures-C-Schaums-Outline/dp/0070701989/ref=sr_1_1?dchild=1&keywords=Data+Structures+with+C+%28Schaum%27s+Outline+Series%29&qid=1593517616&sr=8-1

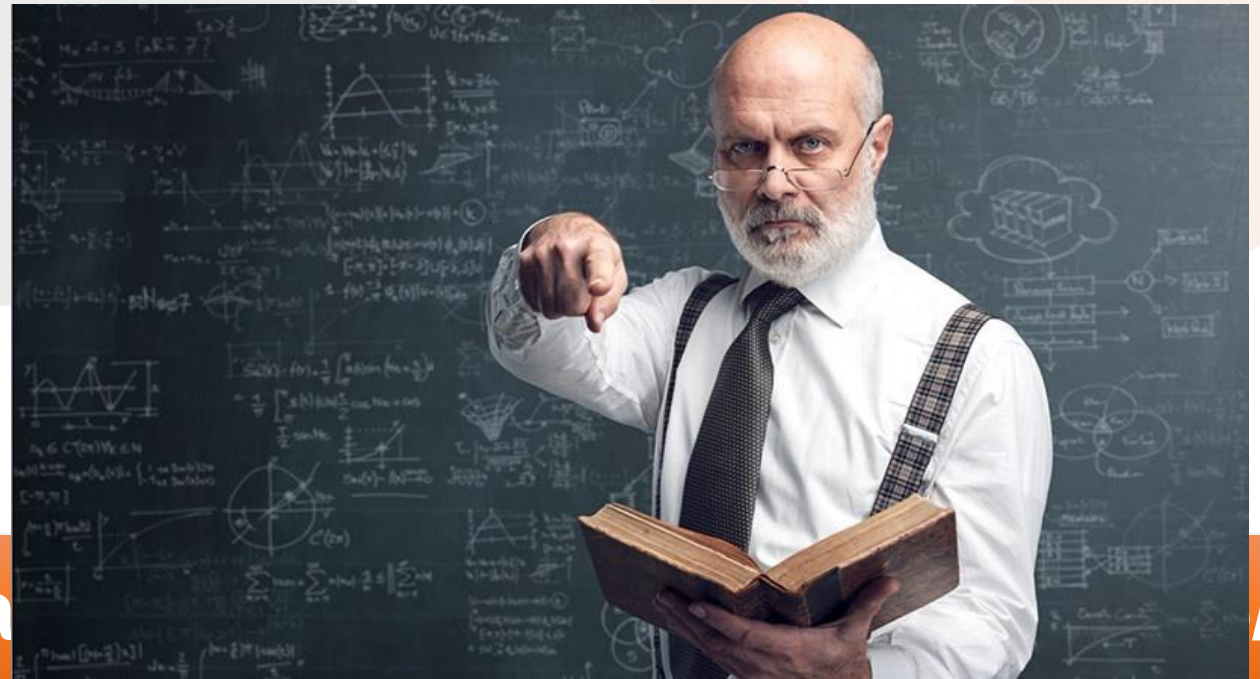
https://www.amazon.in/Fundamentals-Data-Structures-C-Second/dp/8173716056/ref=sr_1_1?dchild=1&keywords=Fundamentals+of+Data+Structures+in+C&qid=1593517838&sr=8-1

What you can expect from me

- Will take care of theory and numerical both
- Will give more weightage to the topics that are asked more frequently in GATE
- Will not emphasize more than required, on a topic
- Will provide PDF of related books and my personal notes

What i expect from you

- There is no hurry, feel free to ask questions any time through out the class, but first listen
- Please revise the entire lecture before and after the class
- Be regular, Consistency is most important
- More you practice, more clarity you will get
- If we follow all the above specified points Success in guaranteed



Break

Idea of computer science

- Computer science deals with solving a problem correctly in the form of Algorithm which then can be converted into a program, in most efficient time and memory.

- To write an efficient program we need knowledge of both Data Structures and Algorithms.
  - ‘DATA STRUCTURE + ALGORITHM = PROGRAM’

Why we study data structure and algorithms

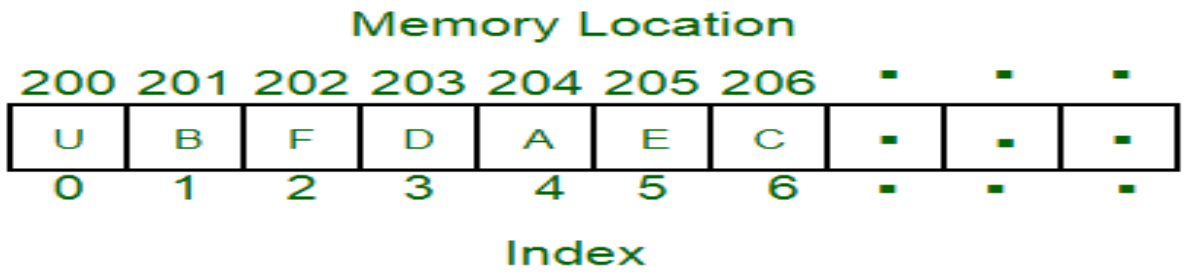
- Course objective is to teach you how to code efficiently.
- What is the meaning of efficiency (time, space, battery, system buses, register etc) time is considered as most important.
- Better running time is obtained from the use of most appropriate data structure and algorithms, rather than through removing a few statements by clever coding.

What is data structure

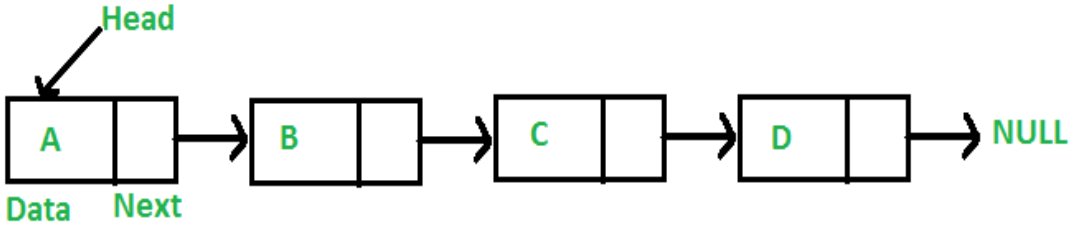
- **Data structure** is a particular way of organizing data in a computer memory (cache, main, secondary) so that Memory can be used efficiently both in terms of time and space.

- It is a logical relationship existing between individual elements of data, it considers elements stored and also their relationship to each other.

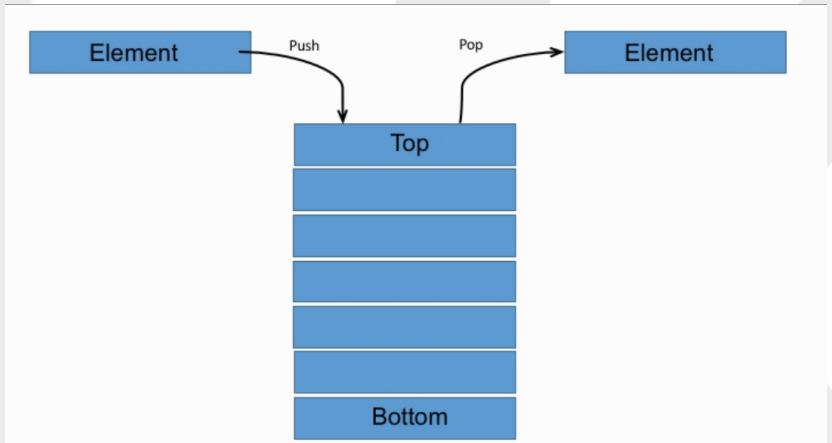
- Data structure mainly specifies the following four things: -
  - Organization of data
  - Accessing methods
  - Degree of association
  - Processing methods



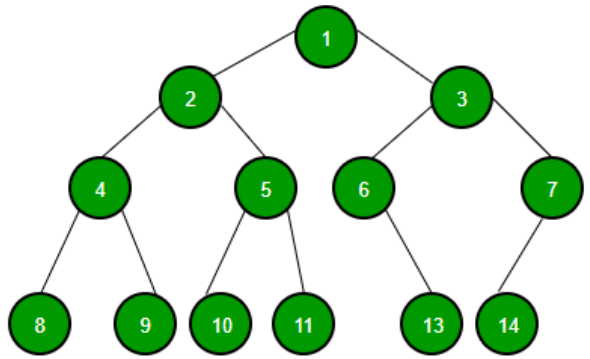
Array



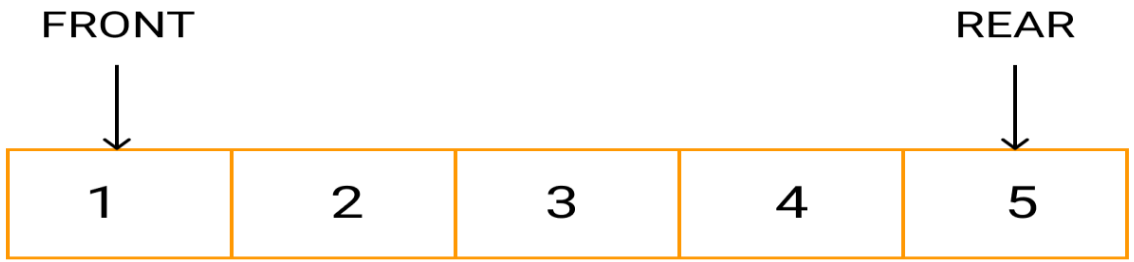
Link List



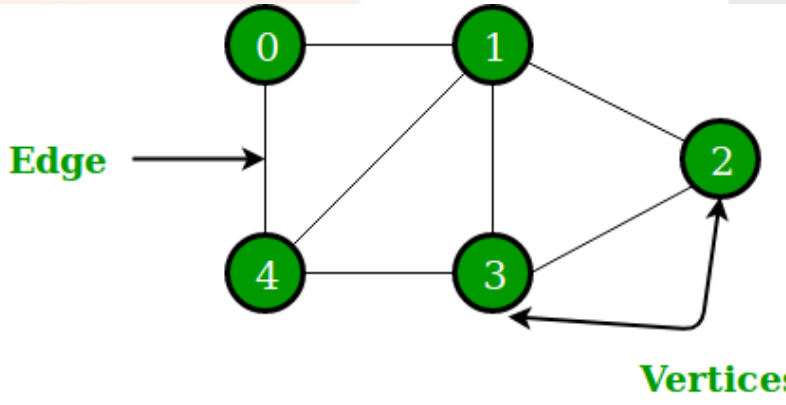
Stack



Tree



Queue

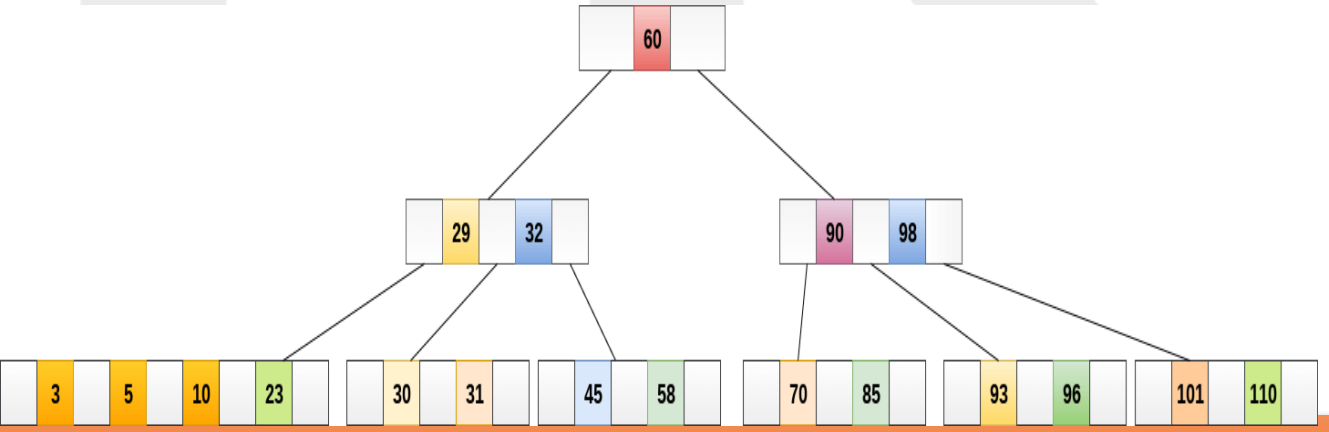


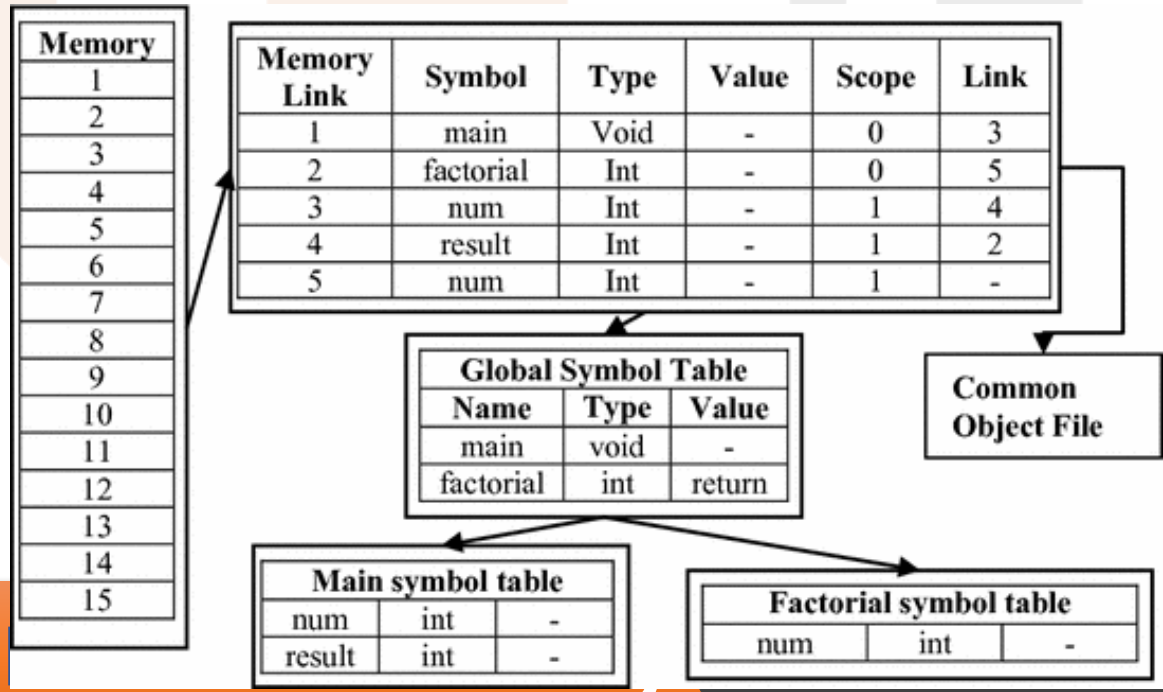
Graph

Effect of Data Structure

- Data structure effect both the structural and functional aspects of the program.

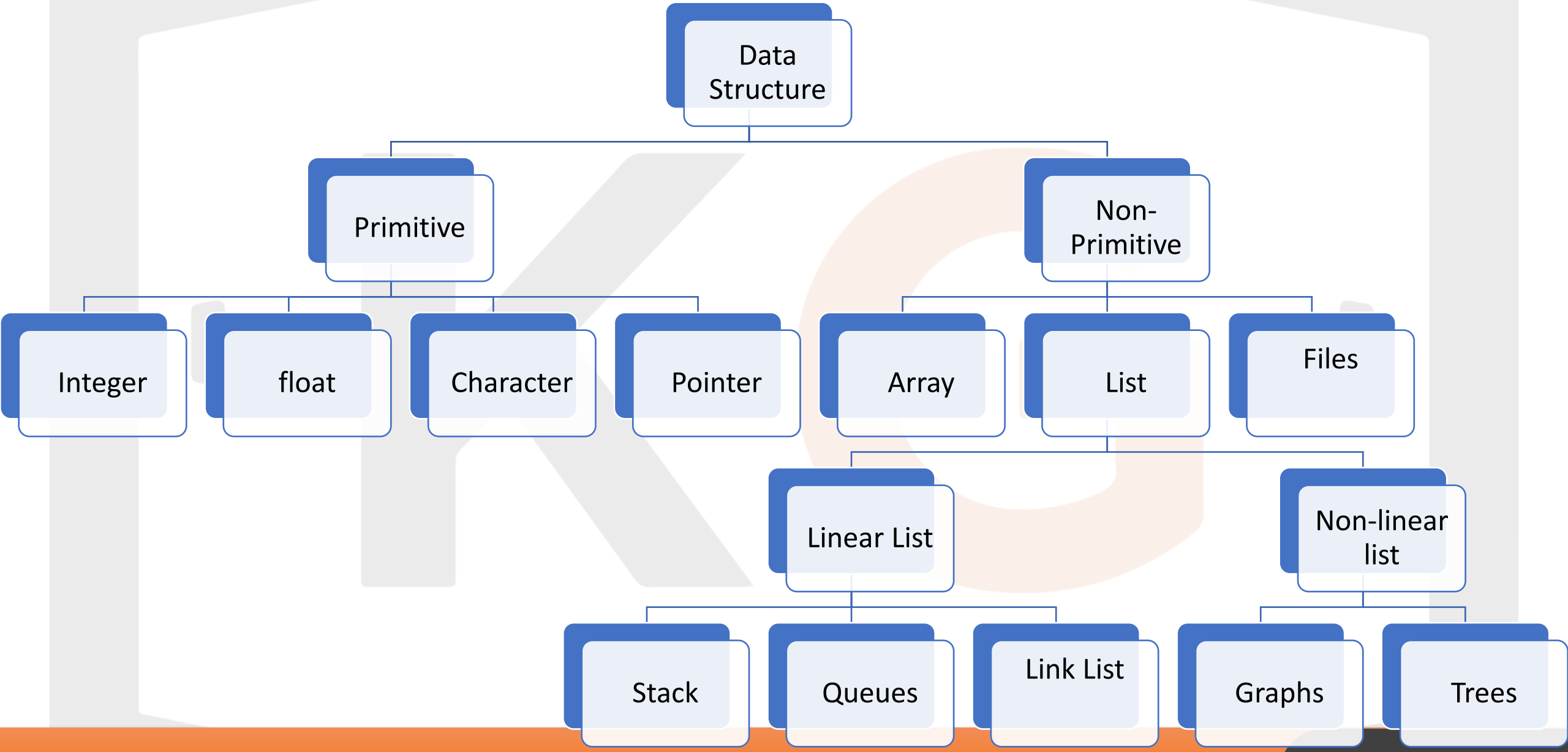
- Different kinds of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks, For example.
  - Relational databases commonly use B-tree indexes for data retrieval
  - Compiler implementations usually use hash tables to look up identifiers.





- Usually, efficient data structures are key to designing efficient algorithms. Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design.

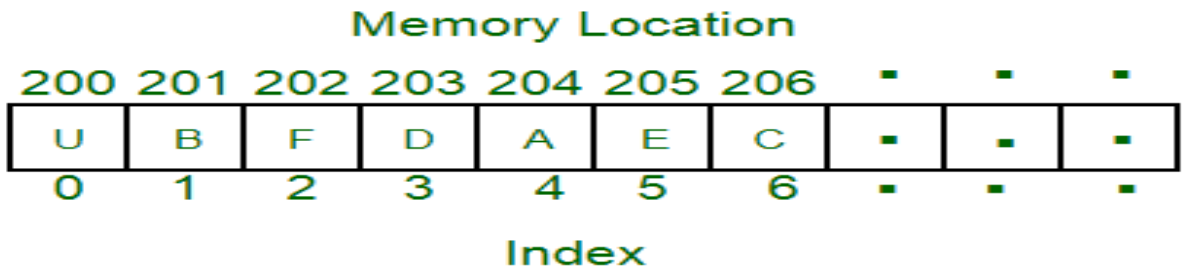
- The implementation of a data structure usually requires writing a set of procedures that create and manipulate instances of that structure.



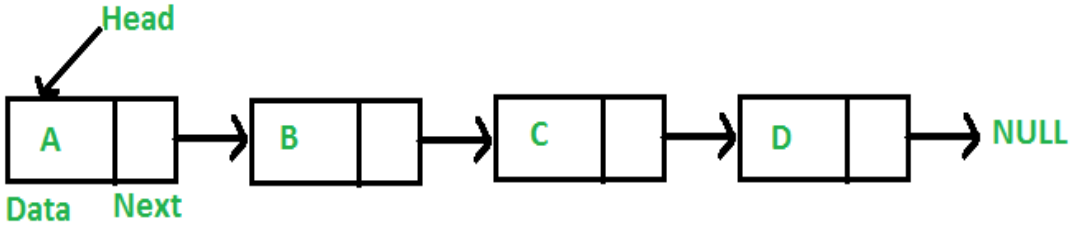
Primitive data structure

- Primitive data structures are those which have predefined way of storing data by the system. And the set of operations that can be performed on these data are also predefined. They are directly operated upon by the machine instruction.

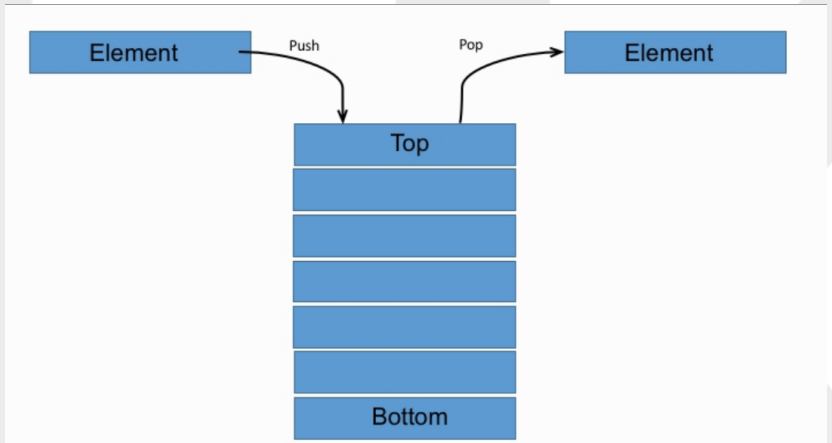
- Primitive data structures are char, int, float, double. The predefined operations are addition, subtraction, etc.



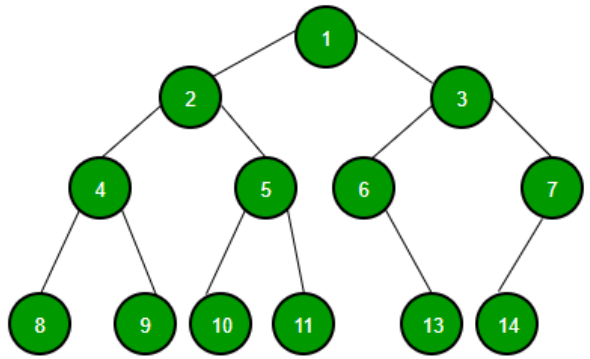
Array



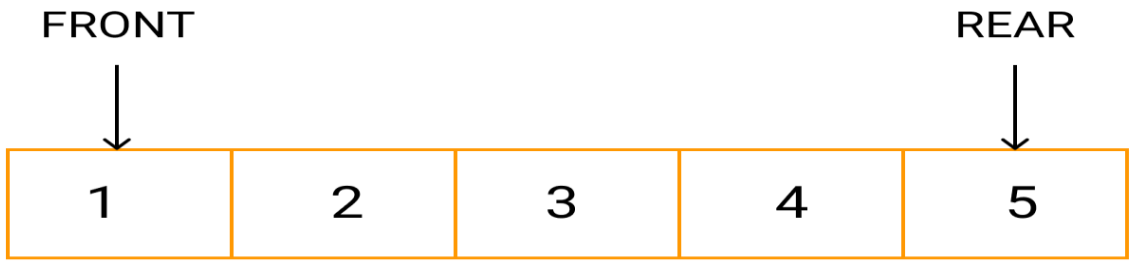
Link List



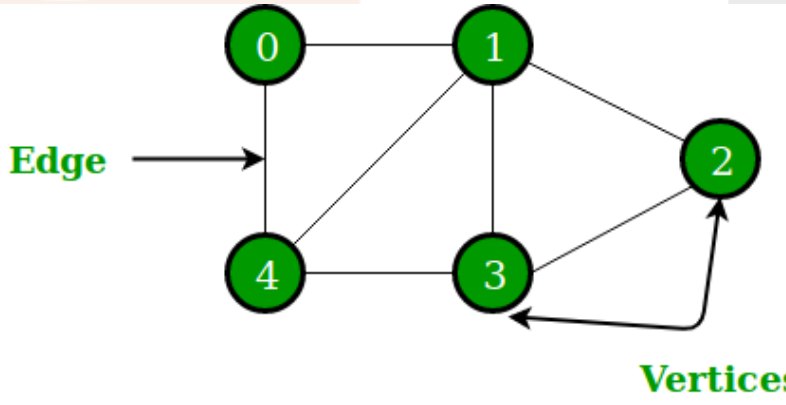
Stack



Tree



Queue



Graph

Non-Primitive data structure

- But there are certain situations when primitive data structures are not sufficient for our job. There comes derived data structures and user defined data structures.

- Derived data structures are also provided by the system but are made using primitives like an array. It can be array of chars, array of int, etc. The set of operations that can be performed on derived data structures are also predefined.

- Finally, there are user defined data types which the user defines using the primitive and derived data types using language constructs like structure or class and uses according to their needs. And the user has to define the set or operations that we can perform on them. User defines data types are Linked Lists, Trees, etc.

LINEAR DATA STRUCTURE	NON-LINEAR DATA STRUCTURE
• In a linear data structure, data elements are arranged in a linear order where each and every element are attached to its previous and next adjacent.	• In a non-linear data structure, data elements are attached in hierarchical manner.
• In linear data structure, single level is involved.	• Whereas in non-linear data structure, multiple levels are involved.
• Its implementation is easy in comparison to non-linear data structure.	• While its implementation is complex in comparison to linear data structure.
• In linear data structure, data elements can be traversed in a single run only.	• While in non-linear data structure, data elements can’t be traversed in a single run only.
• Its examples are: array, stack, queue, linked list, etc.	• While its examples are: trees and graphs.

- ***Homogeneous data*:** Homogeneous data structures are those data structures that contain only similar type of data e.g. like a data structure containing only integer or float values. The simplest example of such type of data structures is an Array.

- ***Heterogeneous Data*:** Heterogeneous Data Structures are those data structures that contains a variety or dissimilar type of data, for e.g. a data structure that can contain various data of different data types like integer, float and character. The examples of such data structures include structures, union etc.

- There are many basic data structures that can be used to solve application problems. Array is a good static data structure that can be accessed randomly and is fairly easy to implement.

- Linked Lists on the other hand is dynamic and is ideal for application that requires frequent operations such as add, delete, and update. One drawback of linked list is that data access is sequential.

- One of the disadvantages of using an array or linked list to store data is the time necessary to search for an item. Since both the arrays and Linked Lists are **linear structures** the time required to search a “linear” list is proportional to the size of the data set.

- One other data structure is the hash table that allows users to program applications that require frequent search and updates. They can be done in O(1) in a hash table.

- Then there are other specialized data structures like, stacks and queues that allows us to solve complicated problems (eg: Maze traversal) using these restricted data structures.