WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

DATA STRUCTURE

ROSHNI SHARMA.P, DIVYASRI.R, KAVYA.S, LATHIKKA RUTHUWA.R

A B S T R A C T

A data structure is a collection of data values and the connections between them. Data structures allow programs to store and reuse data effectively. There are numerous different data structures, and each one of it has its own advantages and disadvantages. Arrays, lists, trees, and graphs are some of the most common data structures.

1.INTRODUCTION

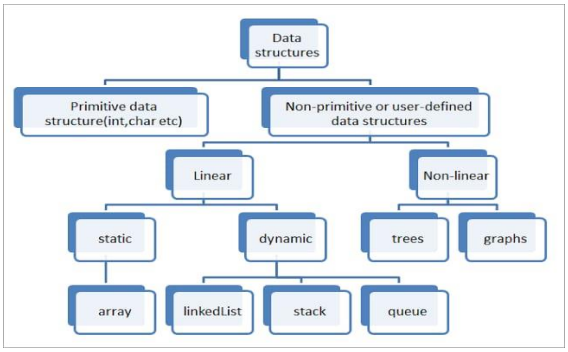
In the environment of computers, the data structure is a specific way of storing and organizing data in the computer's memory so that these data can be fluently recaptured and efficiently used when demanded latterly. The data can be managed in numerous different ways, similar as a logical or fine model for a particular association of data is called a data structure.

DATA STRUCTURE ***CLASSIFICATION***

Types:

Data structures can be classified into two major types:

- ❖ Primitive Data Structures
- ❖ Non Primitive Data Structure



2.LINEAR DATA STRUCTURE

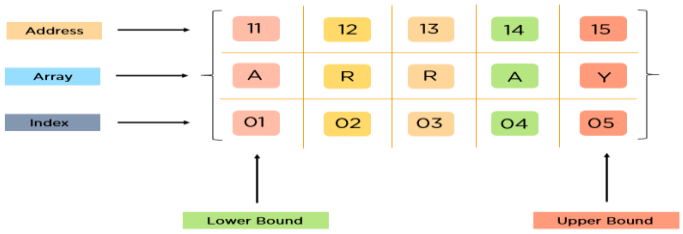
A data structure is said to be direct if its rudiments combine to form any specific order.

There are two ways for representing similar direct structure within memory. The first way is to give a direct relationship between all the rudiments represented using a direct memory position. These direct structures are called arrays. The alternate fashion provides a direct relationship between all the rudiments represented using the conception of pointers or links. These direct structures are called linked lists.

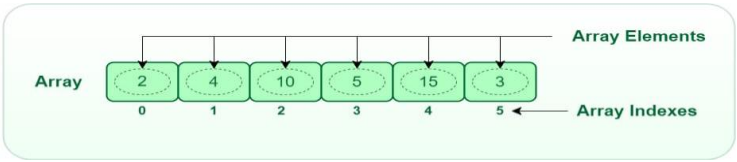
The typical exemplifications of the direct data structure are:

- Arrays
- Ranges
- Stacks
- Linked lists

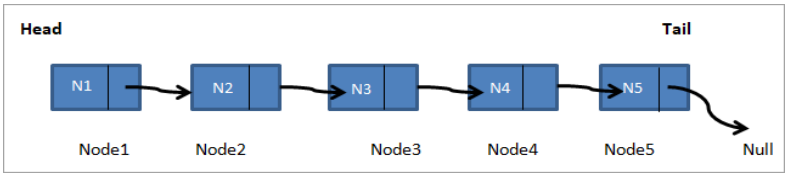
2.1 ARRAY



An array is a collection of analogous data rudiments store at conterminous memory locales. It's the simplest data structure where each data element can be penetrated directly by only using its indicator number.



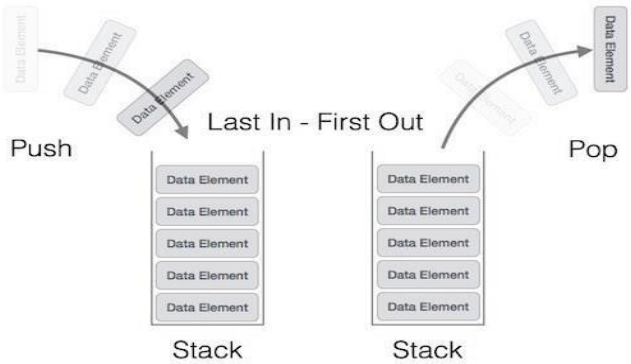
2.2 LINKED LIST



A linked list is a direct data structure that's used to maintain a list- suchlike structure in the computer memory. It's a group of bumps that aren't stored at conterminous locales. Each knot of the list is linked to its conterminous knot with the help of pointers.

2.3STACK AND QUEUE

A stack is a direct data structure that follows a specific order during which the operations are performed.
The order could be FILO(First In Last eschewal) or LIFO(Last In First eschewal).

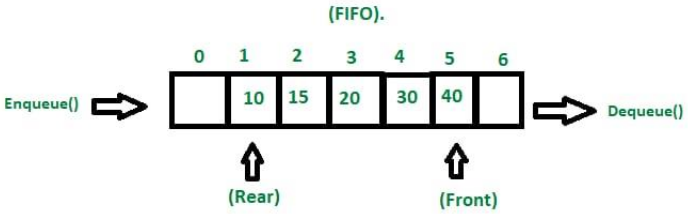


The introductory operations performed in mound are as follows:

❖ Push – Adds an item within the mound.
❖ Pop – Deletes or removes an item from the mound.
❖ Top – Returns the topmost element of the mound.

Queue is a direct data structure in which rudiments can be fitted from only one end which is known as hinder and deleted from another end known as front. FIFO(First In First Out) order is followed by queue.

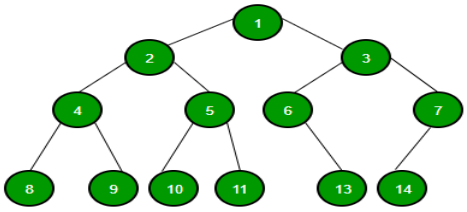
Deque – Adds an element to the line.
Enqueue – Deletes or removes an element from the line.
IsFull – Checks if the queue is full and returns true if the line is full.
IsEmpty – Checks if the queue is empty and returns true if the line is empty.



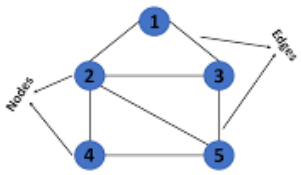
3.NON LINEAR DATA STRUCTURE

3.1 TREE

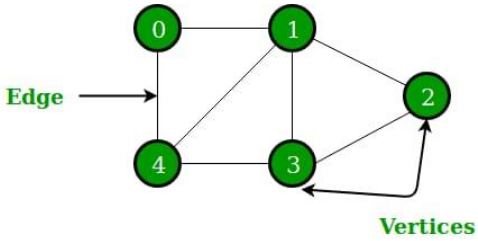
Tree In this case, the data frequently has hierarchical relationship between the different rudiments.
The data structure that represents this relationship is called a confirmed tree graph or tree.



GRAPH



Graph In this case, the data occasionally has connections between dyads of rudiments, which don't inescapably follow a hierarchical structure. This is called a graph.



4. DATA STRUCTURE OPERATIONS

➢Searching – We can fluently search for any data element in a data structure.
➢ Sorting – We can sort the rudiments either in thrusting or descending order.
➢ Insertion – We can fit new data rudiments in the data structure.
➢ Omission – We can cancel the data rudiments from the data structure.
➢ Updation – We can modernize or replace the living rudiments from the data structure..

5. DATA STRUCTURE APPLICATIONS

❖ Representation of information in databases
❖ Algorithms that search through data(analogous as a quest machine)
❖ algorithms that manipulate data(analogous as a word processor)
❖ algorithms that anatomize data(analogous as a data miner)
❖ algorithms that induce data(analogous as a arbitrary number creator)
❖ algorithms that compress and relax data(analogous as a zip avail)
❖ algorithms that reckon and decrypt data(analogous as a security system)
❖ software that manages lines and directories(analogous as a train director)
❖ software that renders plates analogous as a web cybersurfer or 3D rendering.

6. CONCLUSION

- Data structures are an essential concept in computer science that allow for efficient organization, storage, and retrieval of data in a computer program. They provide a way to manage large sets of data and perform operations on that data, making it easier for software developers to build complex software systems.
- There are many different types of data structures available, each with its own advantages and disadvantages. These include arrays, linked lists, stacks, queues, trees, graphs, and hash tables. Understanding data structures and how to use them efficiently is essential for designing high performance software systems. Properly designed data structures can improve the efficiency of algorithms by reducing the number of operations needed to perform a given task.
- Overall, data structures are a fundamental concept in computer science, and a solid understanding of data structures is necessary for anyone who wants to become a proficient software developer or computer scientist.

References

1. https://www.tutorialspoint.com/data_struct ures_algorithms/array_data_structure.html
2. https://everythingcomputerscience.com/disc rete_mathematics/Stacks_and_Queues.html
3. https://www.simplilearn.com/tutorials/data -structure-tutorial/trees-in-data-structure
4. https://towardsdatascience.com/data structures-and-algorithms-journey 80d225cfbbd8
5. https://www.google.com/search?q=articles+ data+structure&rlz=1C1RXQR_enIN1032IN10 32&oq=&aqs=chrome.0.69i59i450l8.19168287j0j15&sourceid=chrome&ie=UTF-8