UNIVERSITY OF ANBAR

Computer Networking Systems Department

Data structure

Daniah Abdul Qahar Shakir

2024-2025

Lecture 1

First course

Data item: an elementary description of things, events, actions, and transactions that are recorded , classified and stored ,but not organized to convey any specific meeting.

Information: is data that been organized so that they have meaning and value to the reception.

Data structure: a collection of data elements whose organization is characterized by accessing (mechanism for organizing information) operation that are used to store and retrieve the individual of elements

Data structure: is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.

Algorithm: it is finite set instruction which, if followed, accomplish a particular task.

Program : an implementation of an algorithm in some programming language such as C,C++,c#, etc..

How to Choose the Right Data Structure for Your Applications

Choosing the best data structure and algorithm for a coding interview is not a trivial task. You need to consider the problem, the input, the output, the constraints, the trade-offs, and the edge cases. Learn some tips and strategies to help you ace your coding interview by picking the right data structure and algorithm. Here's a guide to help you make the right choice

1. data size and required memory
2. the dynamic nature of data.
3. the required time to obtain any data element from the data structure.
4. the programing approach and algorithm that will be used to manipulate these .

Types of Data Structure

Basically, data structures are divided into two categories:

- Linear data structure
- Non-linear data structure

Let's learn about each type in detail.

Linear data structures

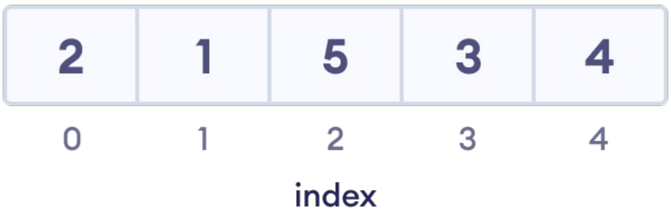
In linear data structures, the elements are arranged in sequence one after the other. Since elements are arranged in particular order, they are easy to implement.

However, when the complexity of the program increases, the linear data structures might not be the best choice because of operational complexities.

Popular linear data structures are:

1. Array Data Structure

In an array, elements in memory are arranged in continuous memory. All the elements of an array are of the same type. And, the type of elements that can be stored in the form of arrays is determined by the programming language.

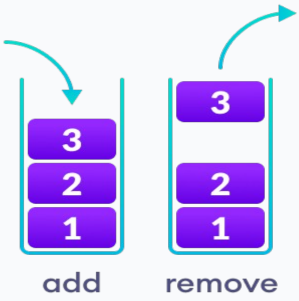


An array with each element represented by an index

2. Stack Data Structure

In stack data structure, elements are stored in the LIFO principle. That is, the last element stored in a stack will be removed first.

It works just like a pile of plates where the last plate kept on the pile will be removed first.

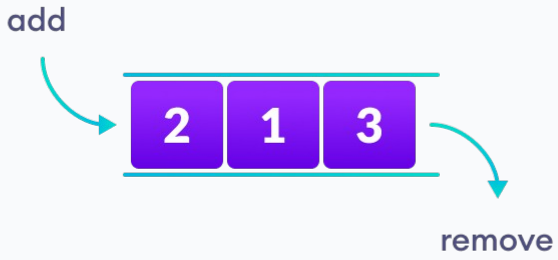


In a stack, operations can be perform only from one end (top here).

3. Queue Data Structure

Unlike stack, the queue data structure works in the FIFO principle where first element stored in the queue will be removed first.

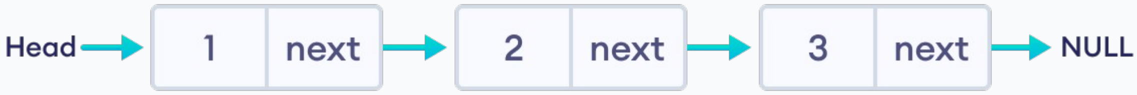
It works just like a queue of people in the ticket counter where first person on the queue will get the ticket first.



In a queue, addition and removal are performed from separate ends.

4. Linked List Data Structure

In linked list data structure, data elements are connected through a series of nodes. And, each node contains the data items and address to the next node.



A linked list

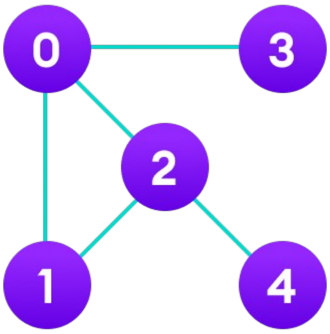
Non linear data structures

Unlike linear data structures, elements in non-linear data structures are not in any sequence. Instead they are arranged in a hierarchical manner where one element will be connected to one or more elements.

Non-linear data structures are further divided into graph and tree based data structures.

1. Graph Data Structure

In graph data structure, each node is called vertex and each vertex is connected to other vertices through edges.



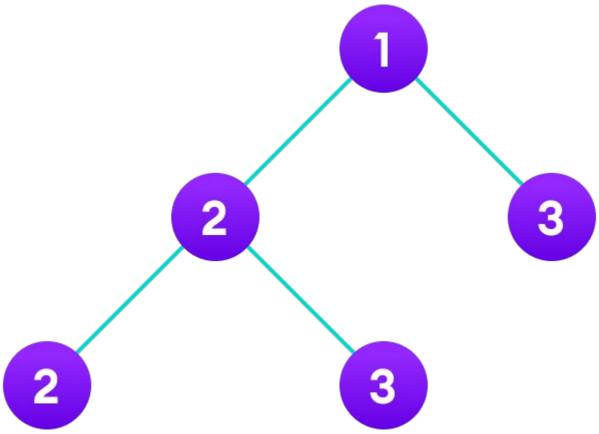
Graph data structure example

Popular Graph Based Data Structures:

- Spanning Tree and Minimum Spanning Tree
- Strongly Connected Components
- Adjacency Matrix
- Adjacency List

2. Trees Data Structure

Similar to a graph, a tree is also a collection of vertices and edges. However, in tree data structure, there can only be one edge between two vertices.



Tree data structure example

Popular Tree based Data Structure

- Binary Tree
- Binary Search Tree

Linear Vs Non-linear Data Structures

Now that we know about linear and non-linear data structures, let's see the major differences between them.

Linear Data Structures	**Non Linear Data Structures**
The data items are arranged in sequential order, one after the other.	The data items are arranged in non-sequential order (hierarchical manner).
All the items are present on the single layer.	The data items are present at different layers.
It can be traversed on a single run. That is, if we start from the first element, we can traverse all the elements sequentially in a single pass.	It requires multiple runs. That is, if we start from the first element it might not be possible to traverse all the elements in a single pass.
The memory utilization is not efficient.	Different structures utilize memory in different efficient ways depending on the need.
The time complexity increase with the data size.	Time complexity remains the same.
Example: Arrays, Stack, Queue	Example: Tree, Graph, Map

References

- Introduction to Algorithms, 3rd Edition by *Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein*

- Introduction to Algorithms, 3rd Edition by *Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein*

- Elements of Programming Interviews in Java: The Insiders' Guide, by *Adnan Aziz, Tsung-Hsien Lee, Amit Prakash*

- https://github.com/careermonk/DataStructuresAndAlgorithmsMadeEasy