

Join our newsletter for the latest updates. Enter Email Add Subscribe

Data Structure and Types

In this article, you will learn about data strucrture and its types.

What are Data Structures?

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.

Depending on your requirement and project, it is important to choose the right data structure for your project. For example, if you want to store data sequentially in the memory, then you can go for the Array data structure.



Basically, data structures are divided into two categories:

Types of Data Structure

Linear data structure

- Non-linear data structure
- Let's learn about each type in detail.

Since elements are arranged in particular order, they are easy to implement.

Linear data structures

However, when the complexity of the program increases, the linear data structures

In linear data structures, the elements are arranged in sequence one after the other.

Popular linear data structures are: 1. Array Data Structure

might not be the best choice because of operational complexities.

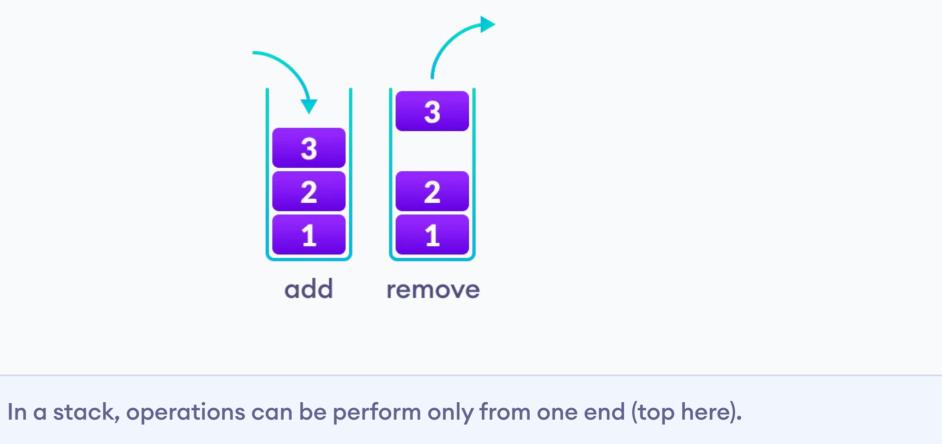
In an array, elements in memory are arranged in continuous memory. All the elements

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. To learn more, visit Java Array.



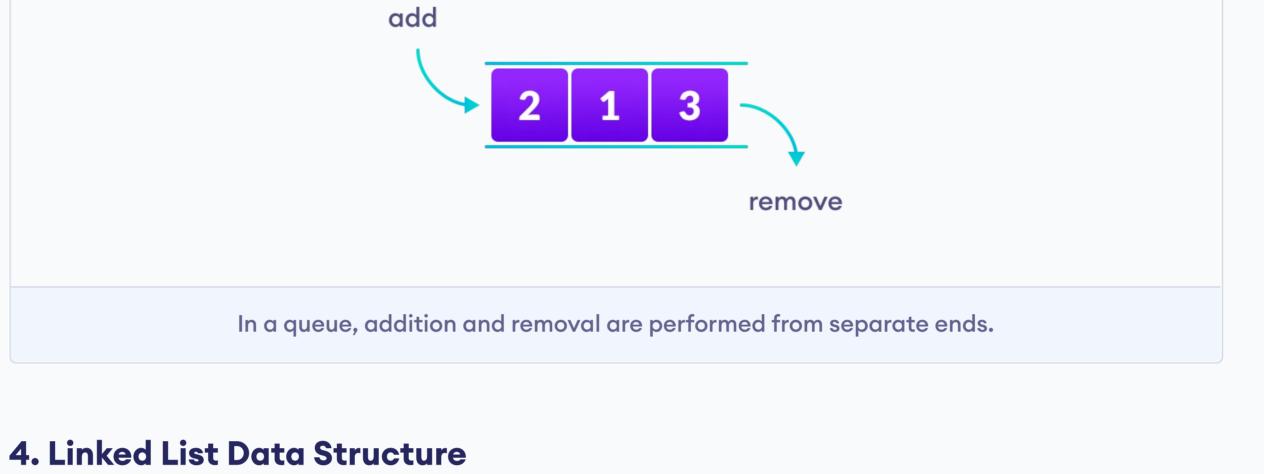
element stored in a stack will be removed first. It works just like a pile of plates where the last plate kept on the pile will be removed first. To learn more, visit Stack Data Structure.



3. Queue Data Structure Unlike stack, the queue data structure works in the FIFO principle where first element

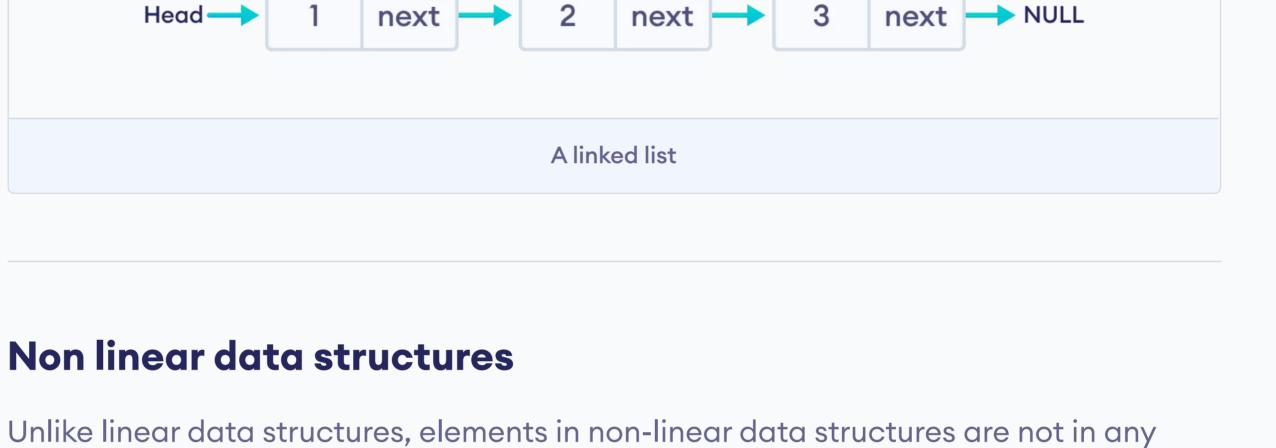
stored in the queue will be removed first.

It works just like a queue of people in the ticket counter where first person on the queue will get the ticket first. To learn more, visit Queue 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.

To learn more, visit Linked List Data Structure.

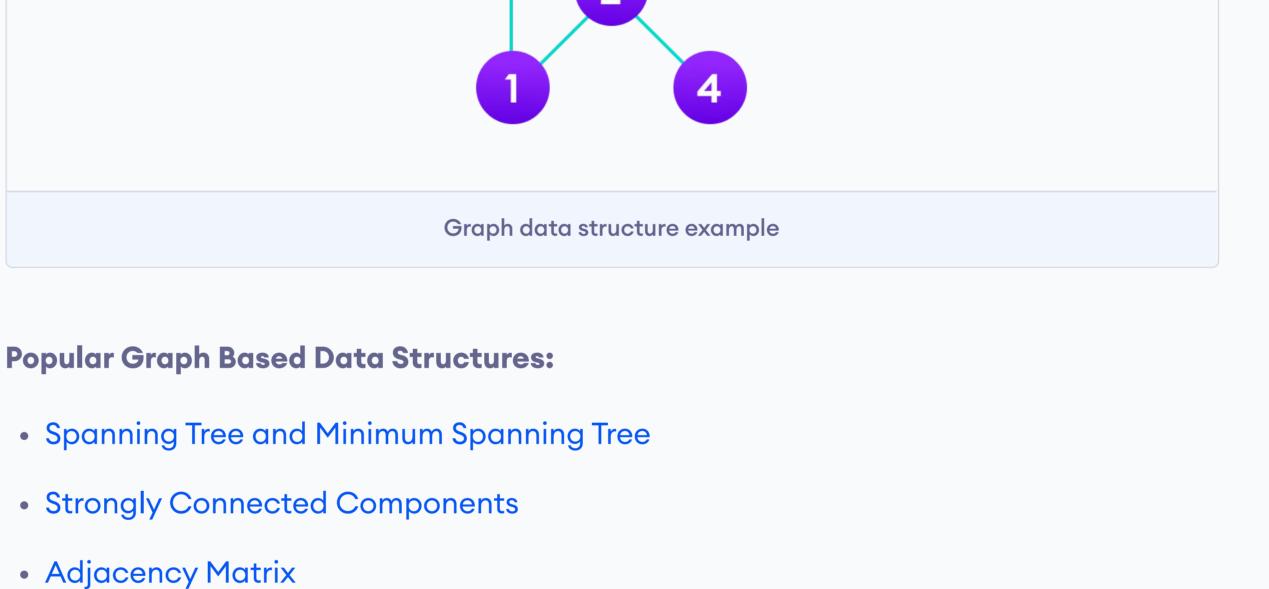


sequence. Instead they are arranged in a hierarchical manner where one element will be connected to one or more elements.

Non-linear data structures are further divided into graph and tree based data structures. 1. Graph Data Structure

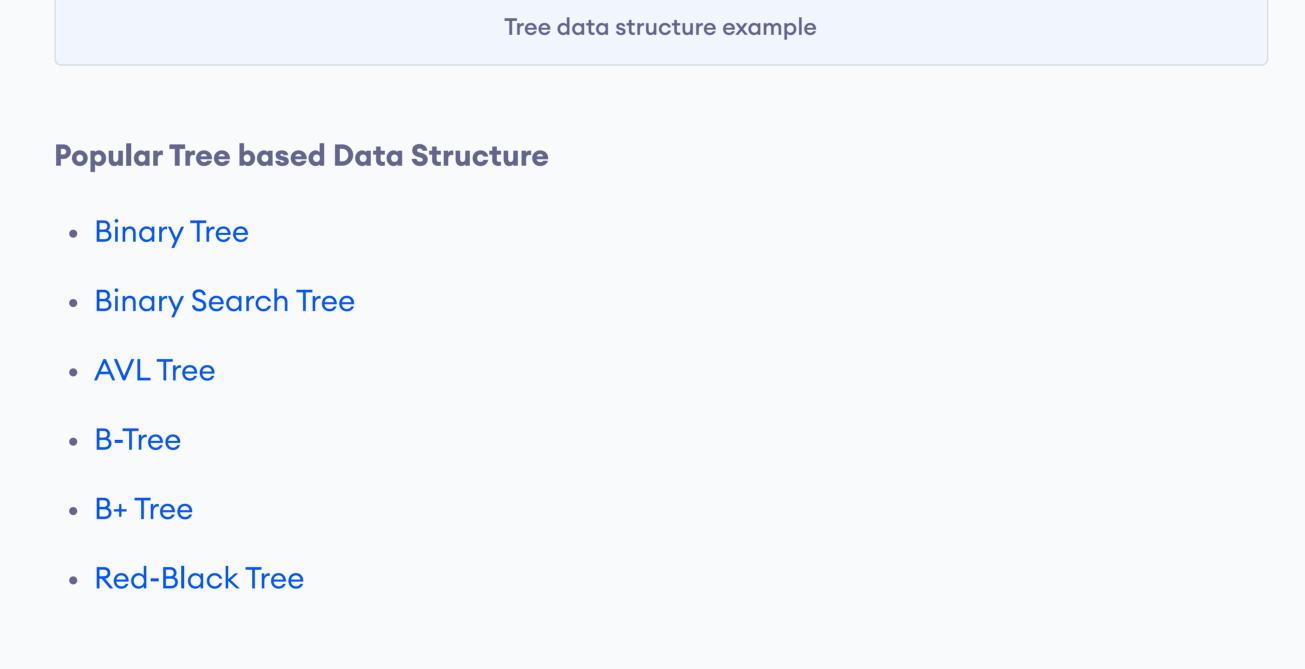
In graph data structure, each node is called vertex and each vertex is connected to

other vertices through edges. To learn more, visit Graph Data Structure.



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. To learn more, visit Tree Data Structure.



It can be traversed on a single run. That is, It requires multiple runs. That is, if we if we start from the first element, we can traverse all the elements sequentially in a

The data items are arranged in sequential

All the items are present on the single

Linear Vs Non-linear Data Structures

differences between them.

Linear Data Structures

order, one after the other.

layer.

Share on:

start from the first element it might not be possible to traverse all the elements in a single pass. single pass.

layers.

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

Non Linear Data Structures

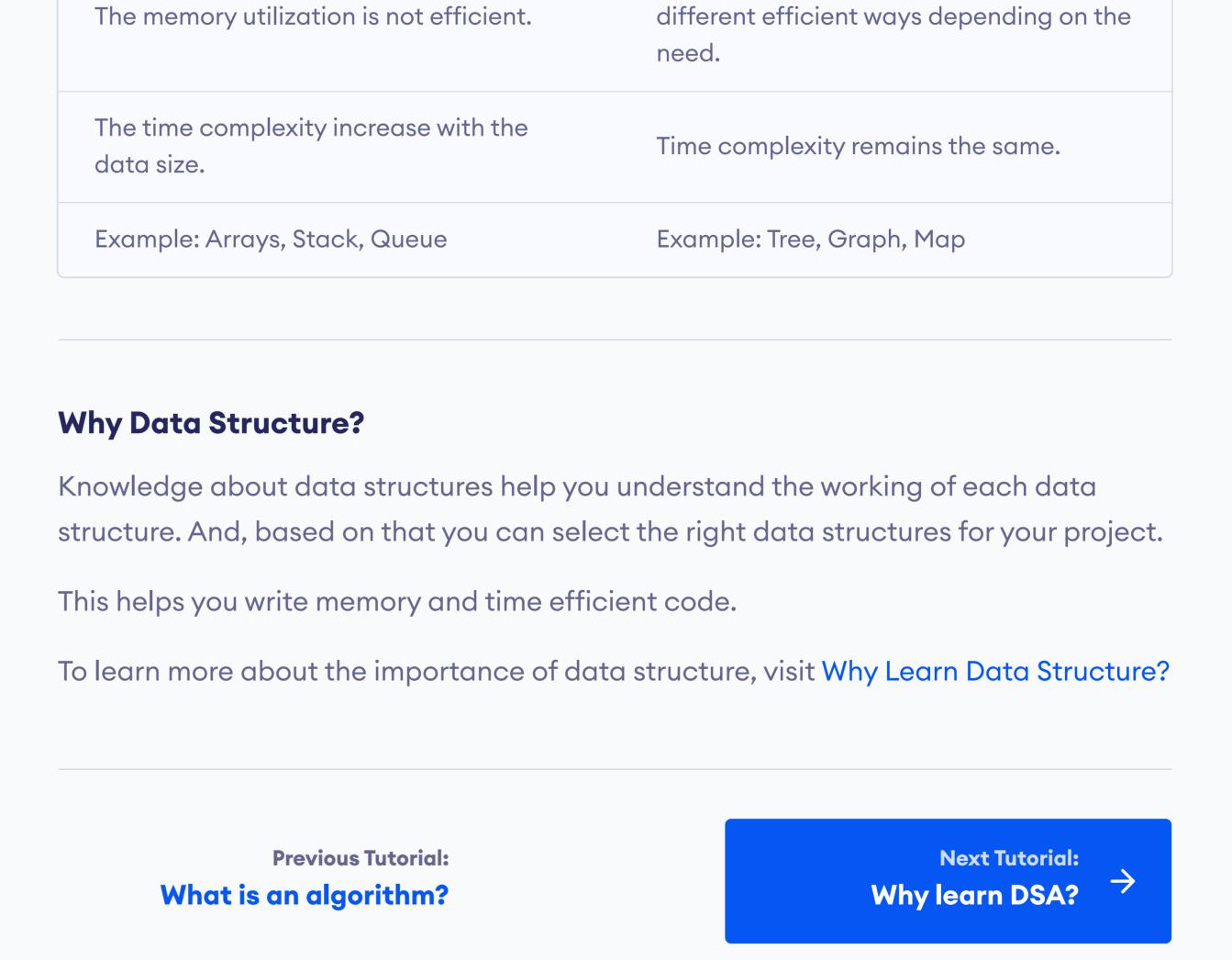
The data items are arranged in non-

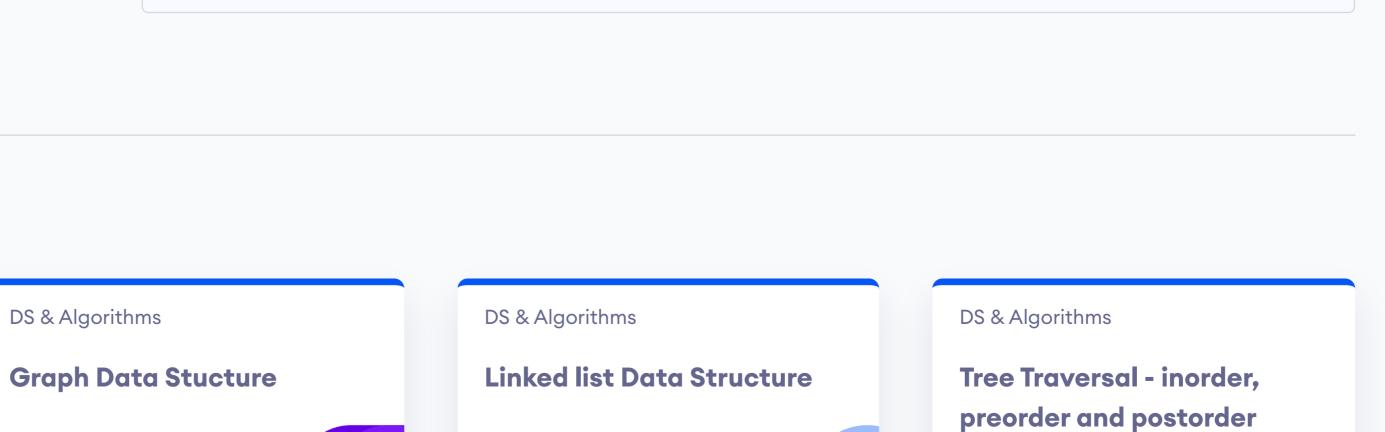
sequential order (hierarchical manner).

The data items are present at different

Different structures utilize memory in

Did you find this article helpful?



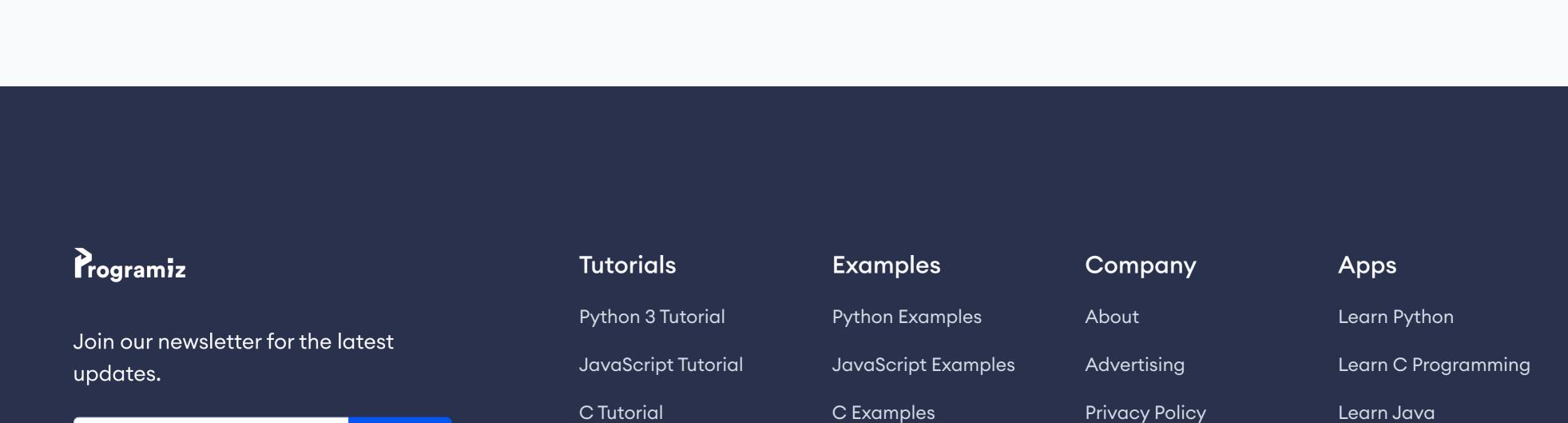


Terms & Conditions

Contact

Youtube

Blog



Java Examples

Kotlin Examples

C++ Examples

Java Tutorial

Kotlin Tutorial

C++ Tutorial

Swift Tutorial

C# Tutorial

DSA Tutorial

© Parewa Labs Pvt. Ltd. All rights reserved.

Related Tutorials

Tree Data Structure

Enter Email Address*

App Store

Join

DS & Algorithms

0