



Graph and its representations

Last Updated : 02 Jan, 2024

What is Graph Data Structure?

A **Graph** is a non-linear data structure consisting of vertices and edges. The vertices are sometimes also referred to as nodes and the edges are lines or arcs that connect any two nodes in the graph. More formally a Graph is composed of a set of vertices(V) and a set of edges(E). The graph is denoted by $G(V, E)$.

Representations of Graph

Here are the two most common ways to represent a graph :

1. Adjacency Matrix
2. Adjacency List

Adjacency Matrix

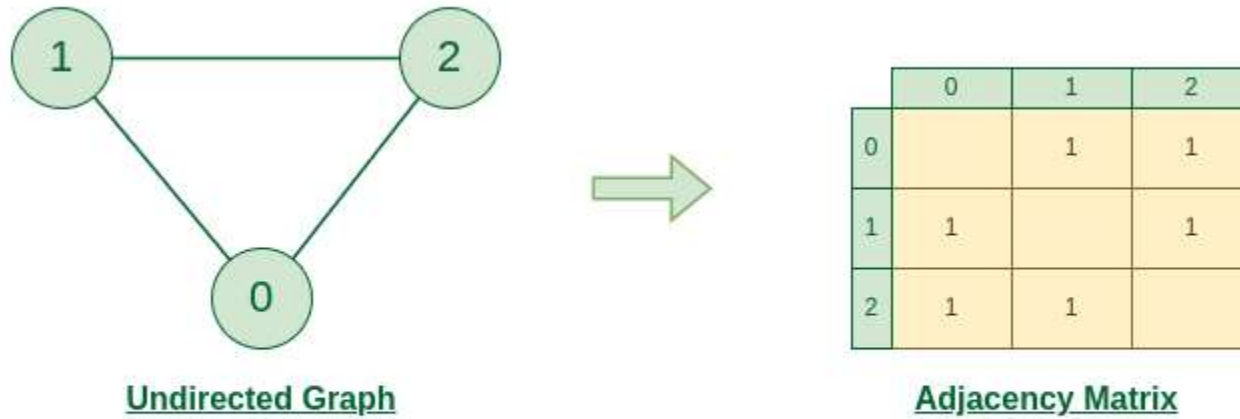
An adjacency matrix is a way of representing a graph as a matrix of boolean (0's and 1's).

Let's assume there are n vertices in the graph So, create a 2D matrix **adjMat[n][n]** having dimension $n \times n$.

- *If there is an edge from vertex i to j , mark $adjMat[i][j]$ as **1**.*
- *If there is no edge from vertex i to j , mark $adjMat[i][j]$ as **0**.*

Representation of Undirected Graph to Adjacency Matrix:

The below figure shows an undirected graph. Initially, the entire Matrix is initialized to **0**. If there is an edge from source to destination, we insert **1** to both cases (**$adjMat[source]$** and **$adjMat[destination]$**) because we can go either way.

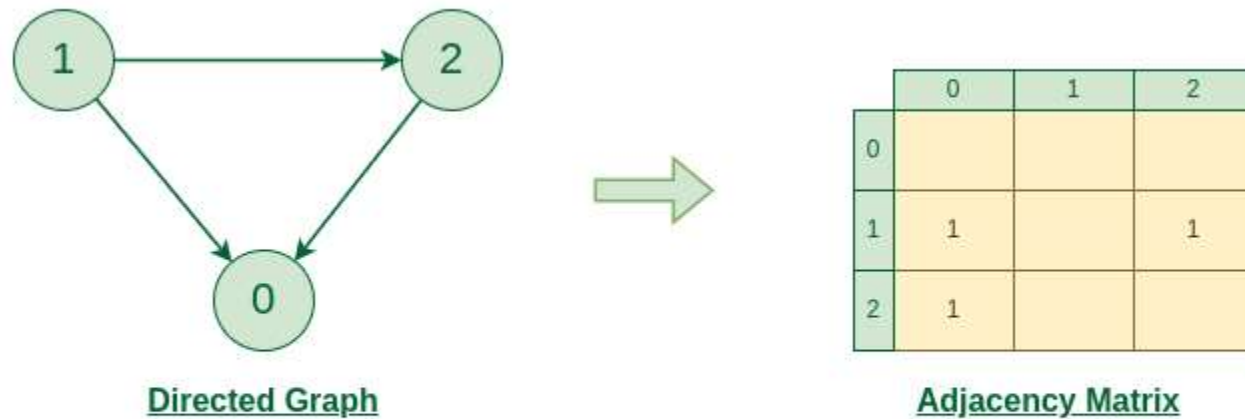


Graph Representation of Undirected graph to Adjacency Matrix

Undirected Graph to Adjacency Matrix

Representation of Directed Graph to Adjacency Matrix:

The below figure shows a directed graph. Initially, the entire Matrix is initialized to **0**. If there is an edge from source to destination, we insert **1** for that particular **adjMat[destination]**.



Graph Representation of Directed graph to Adjacency Matrix

Directed Graph to Adjacency Matrix

Adjacency List

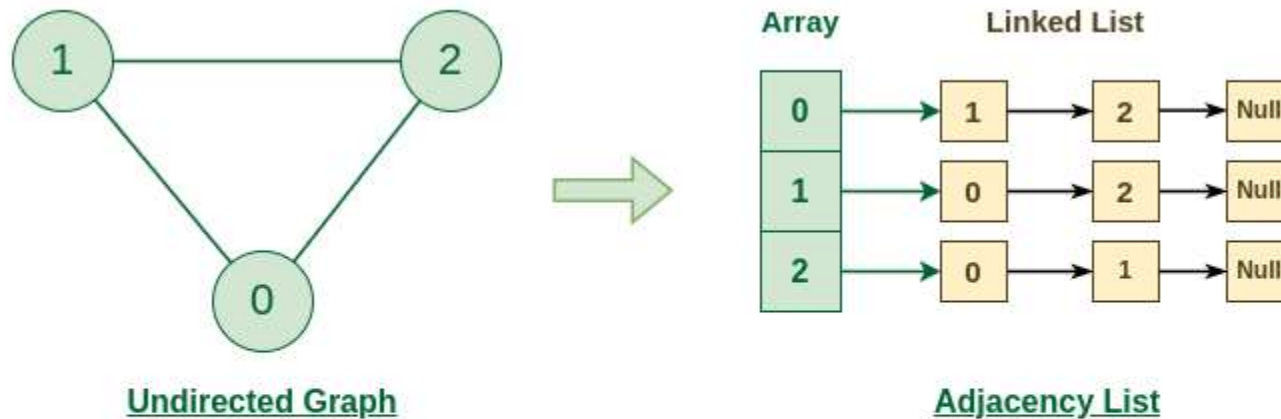
An array of Lists is used to store edges between two vertices. The size of array is equal to the number of **vertices (i.e, n)**. Each index in this array represents a specific vertex in the graph. The entry at the index *i* of the array contains a linked list containing the vertices that are adjacent to vertex *i*.

Let's assume there are **n** vertices in the graph So, create an **array of list** of size **n** as **adjList[n]**.

- *adjList[0] will have all the nodes which are connected (neighbour) to vertex 0.*
- *adjList[1] will have all the nodes which are connected (neighbour) to vertex 1 and so on.*

Representation of Undirected Graph to Adjacency list:

The below undirected graph has 3 vertices. So, an array of list will be created of size 3, where each indices represent the vertices. Now, vertex 0 has two neighbours (i.e, 1 and 2). So, insert vertex 1 and 2 at indices 0 of array. Similarly, For vertex 1, it has two neighbour (i.e, 2 and 0) So, insert vertices 2 and 0 at indices 1 of array. Similarly, for vertex 2, insert its neighbours in array of list.



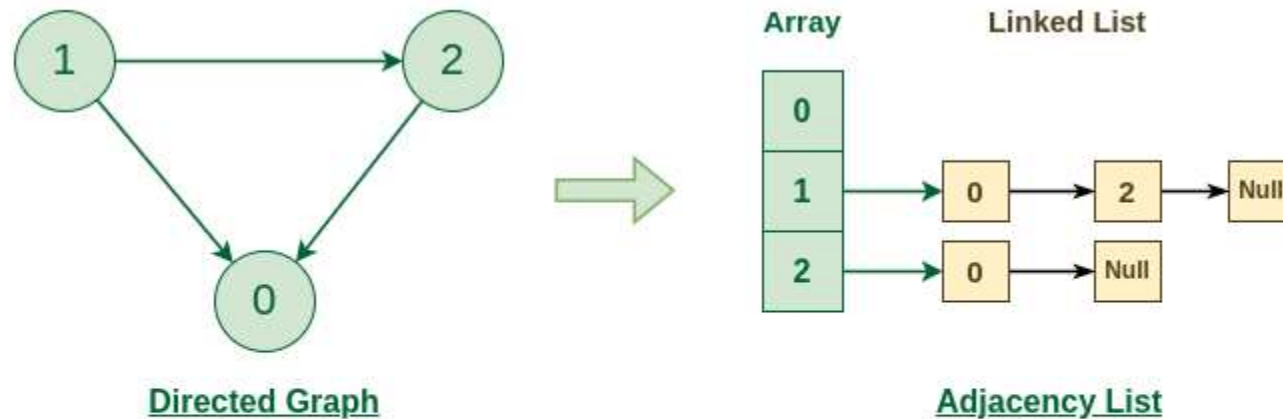
Graph Representation of Undirected graph to Adjacency List

Undirected Graph to Adjacency list

Representation of Directed Graph to Adjacency list:

The below directed graph has 3 vertices. So, an array of list will be created of size 3, where each indices represent the vertices. Now, vertex 0 has no neighbours. For vertex 1, it has two neighbour

(i.e, 0 and 2) So, insert vertices 0 and 2 at indices 1 of array. Similarly, for vertex 2, insert its neighbours in array of list.



Graph Representation of Directed graph to Adjacency List

Directed Graph to Adjacency list

"The DSA course helped me a lot in clearing the interview rounds. It was really very helpful in setting a strong foundation for my problem-solving skills. Really a great investment, the passion Sandeep sir has towards DSA/teaching is what made the huge difference." - **Gaurav | Placed at Amazon**

Before you move on to the world of development, **master the fundamentals of DSA** on which every advanced algorithm is built upon. Choose your preferred language and start learning today:

[DSA In JAVA/C++](#)

[DSA In Python](#)

[DSA In JavaScript](#)

Trusted by Millions, Taught by One- Join the best DSA Course Today!

Recommended Problems

Frequently asked DSA Problems

851

Previous

Introduction to Graphs - Data Structure and Algorithm Tutorials

Share your thoughts in the comments

Similar Reads

Solve Problems

[Suggest improvement](#)

Next

Types of Graphs with Examples

Add Your Comment

Sparse Matrix and its representations | Set 1 (Using Arrays and Linked Lists)

Sparse Matrix and its representations | Set 2 (Using List of Lists and Dictionary of keys)

Graph representations using set and hash

Maximize difference between odd and even indexed array elements by swapping unequal adjacent bits in their binary representations

Maximize difference between odd and even-indexed array elements by rotating their binary representations

XOR of two numbers after making length of their binary representations equal

Digits whose alphabetic representations are jumbled in a given string

Check if binary representations of 0 to N are present as substrings in given binary string

Find the number obtained by concatenating binary representations of all numbers up to N

Rearrange array to make decimal equivalents of reversed binary representations of array elements sorted



GeeksforGeeks

Article Tags : [graph-basics](#) , [DSA](#) , [Graph](#)

Practice Tags : [Graph](#)



A-143, 9th Floor, Sovereign Corporate
Tower, Sector-136, Noida, Uttar Pradesh -
201305



Company

About Us
Legal
Careers
In Media
Contact Us
Advertise with us

Explore

Job-A-Thon Hiring
Challenge
Hack-A-Thon
GfG Weekly Contest
Offline Classes
(Delhi/NCR)
DSA in JAVA/C++

Languages

Python
Java
C++
PHP
GoLang
SQL

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
DSA Interview
Questions

Data Science & ML

Data Science With
Python
Data Science For
Beginner
Machine Learning
Tutorial

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS

GFG Corporate Solution	Master System Design	R Language	Competitive Programming	ML Maths	NextJS
Placement Training Program	Master CP	Android Tutorial		Data Visualisation Tutorial	NodeJs
	GeeksforGeeks Videos			Pandas Tutorial	Bootstrap
	Geeks Community			NumPy Tutorial	Tailwind CSS
				NLP Tutorial	
				Deep Learning Tutorial	
Python Tutorial	Computer Science	DevOps	System Design	School Subjects	Commerce
Python Programming Examples	GATE CS Notes	Git	High Level Design	Mathematics	Accountancy
Django Tutorial	Operating Systems	AWS	Low Level Design	Physics	Business Studies
Python Projects	Computer Network	Docker	UML Diagrams	Chemistry	Economics
Python Tkinter	Database	Kubernetes	Interview Guide	Biology	Management
Web Scraping	Management System	Azure	Design Patterns	Social Science	HR Management
OpenCV Tutorial	Software Engineering	GCP	OOAD	English Grammar	Finance
Python Interview Question	Digital Logic Design	DevOps Roadmap	System Design Bootcamp		Income Tax
	Engineering Maths		Interview Questions		
UPSC Study Material	Preparation Corner	Competitive Exams	More Tutorials	Free Online Tools	Write & Earn
Polity Notes	Company-Wise	JEE Advanced	Software Development	Typing Test	Write an Article
Geography Notes	Recruitment Process	UGC NET	Software Testing	Image Editor	Improve an Article
History Notes	Resume Templates	SSC CGL	Product Management	Code Formatters	Pick Topics to Write
				Code Converters	

Science and Technology Notes	Aptitude Preparation	SBI PO	Project Management	Currency Converter	Share your Experiences
Economy Notes	Puzzles	SBI Clerk	Linux	Random Number Generator	Internships
Ethics Notes	Company-Wise Preparation	IBPS PO	Excel	Random Password Generator	
Previous Year Papers	Companies	IBPS Clerk	All Cheat Sheets		
	Colleges				

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved