# Introduction to Graphs | Graph Data Structure

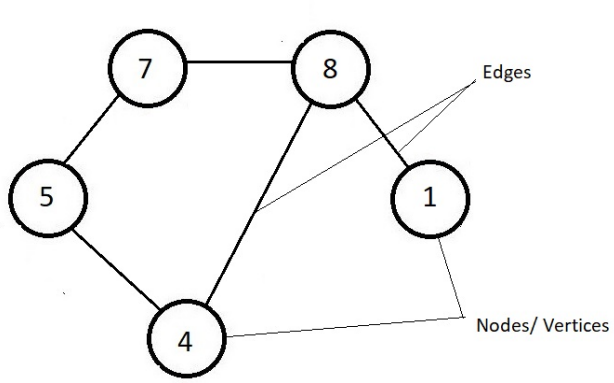
codewithharry.com/videos/data-structures-and-algorithms-in-hindi-83

As we continue on to our topic of data structures and algorithms, we'll examine one of the most important subjects of all, **graphs**. Graphs are an important data structure we should learn. Today, we'll start with the introduction of graphs, and the reasons why we should learn them.

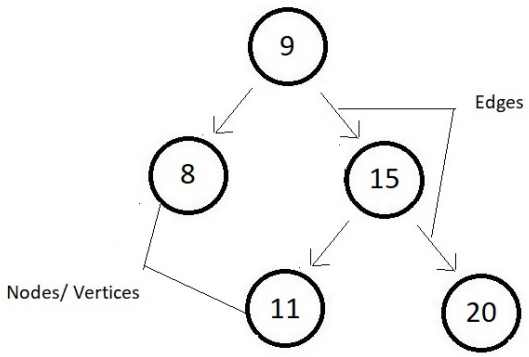
**What is a graph?**

All the data structures we have learned so far have either been linear data structures or nonlinear data structures. Linear data structures included arrays, linked lists, and stacks, while nonlinear hierarchical data structures included Binary Search Trees and AVL Trees. Graphs are an example of non-linear data structures. A graph is a collection of nodes connected through edges.

Example of a simple graph:



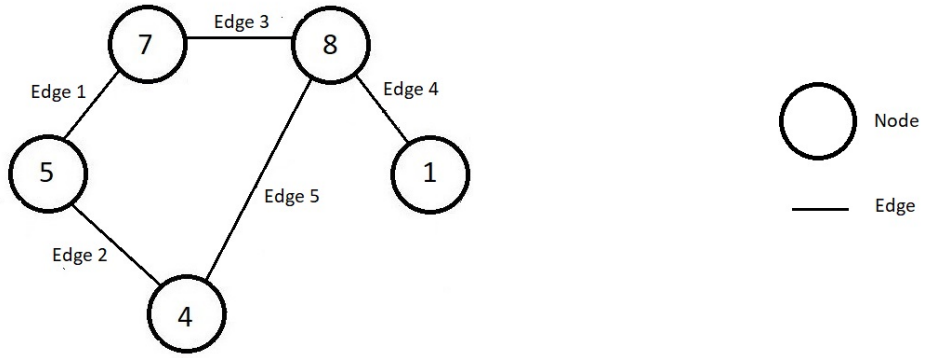

This was one of the general types of graphs we would learn. You might be surprised to learn that trees are also a type of graph. Well, there is nothing as such to be surprised about. Follow the illustration of a tree below.




A tree is also a collection of nodes and edges, and hence is a graph only. Let's now quickly see all those elements of a graph.

## Nodes/ Vertices & Edges:

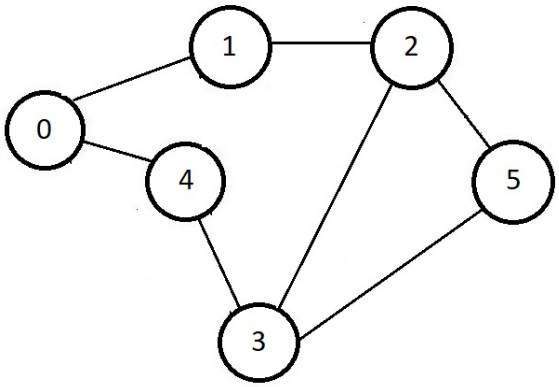
A vertex or node is one fundamental unit/entity of which graphs are formed. Nodes are the data containing parts, connected with each other using edges. An edge is uniquely defined by its 2 endpoints or the two nodes it is connecting. You can take the analogy of people in a network, or a widespread chain, where people could be representing the nodes and their connections in the network could be represented using edges. It is a structure.




## Formal Definition:

A graph G = (V, E) is technically a collection of vertices and edges connecting these vertices.

Follow the below-illustrated graph for reference:




Here, V is the set of all the vertices. Therefore V = {0, 1, 2, 3, 4, 5}, and E is the set of all edges, therefore E = {(0, 1), (1, 2), (0, 4), (4, 3), (3, 5), (2, 5), (2, 3)}. Every edge connects the pair they are represented with. Edge (0, 1) connects node 0 to node 1. And hence, a graph is always represented using its set of vertices, V and its set of edges, E in the form G = (V, E).

## Applications of graphs:

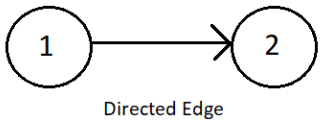
Graphs have a wide range of applications in our world. Learning graphs also becomes a necessity for anyone pursuing software development due to its applications. Graphs are used to model paths in a city, as often seen in Google Maps. They are used to model social

networks such as Facebook or LinkedIn. Graphs are also used to monitor website backlinks, internal employee networks, etc.

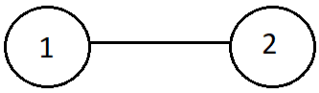
**Types of Edges:**

1. Directed Edge -A directed edge is an edge connecting two nodes but strictly defines the way it is connected from and to. Below example shows node 1 connecting to node 2, and not vice - versa. You can take the analogy of Facebook's follow feature, where if you follow someone, then it's not that the other person who followed you too. It's just one way. Another great example is that of a hyperlink, where one can even link google.com to their own websites on the internet, but then google.com would not necessarily link your website to their page :)

2. Undirected Edge - An undirected edge is an edge connecting two nodes from the way. Below example shows node 1 connecting to node 2, and at the same time node 2 connecting to node 1. You can take the analogy of Facebook's friend feature, where if you make someone a friend, then you too become their friend, so it's both ways.



Directed Edge

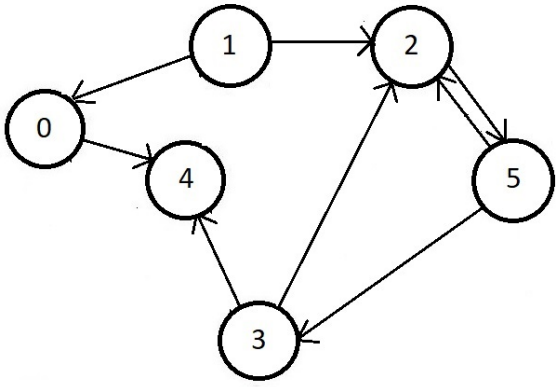


Undirected edge

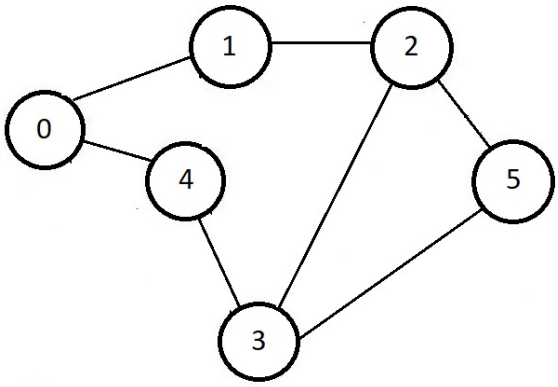
According to the type of edges that a graph has, graphs can be further divided into two types.

**Types of Graphs:**

1. Directed graph - Directed graphs are graphs having each of its edges directed. One of the examples of directed graphs is:



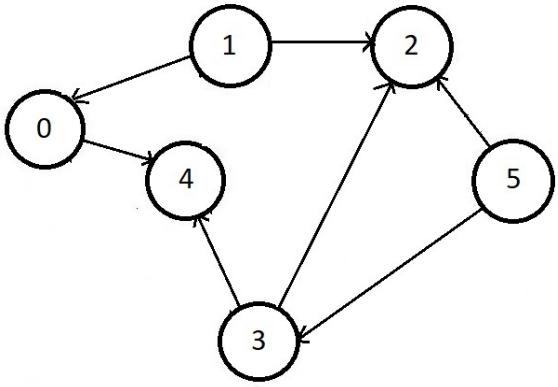

2. Undirected graph - Directed graphs are graphs having each of its edges undirected. One of the examples of directed graphs is:



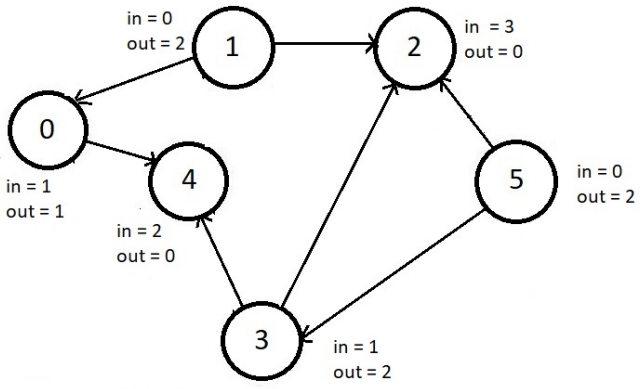

You may sometimes get to see both directed and undirected edges in the same graph, but those are rarely studied. For now, we would restrict ourselves to directed and undirected graphs only.

**Indegree and Outdegree of a node:**

As their name suggests, indegree of a node is the number of edges coming to the node, and outdegree of a node is the number of edges originating from that node. Consider the directed graph below:




We can write the indegree represented by *in* and outdegree represented by *out*, in the above graph for each of these nodes.




Let’s talk about one real-life example of graphs - a graph of users - FACEBOOK!

- Although the users using Facebook need not understand the graph theory, once you create your profile there, Facebook uses graphs to model relationships between nodes.
- We can apply graph algorithms to suggest friends to people, calculate the number of mutual friends, etc.

Suppose you have friends X, Y, and Z on Facebook. And you are one common friend to all of the three. Now, Facebook observes the connections and would suggest your friends to get connected with each other seeing your connection with them. And you would be shown as one mutual friend to all three of them, and this is the concept behind the mutual friends and friends suggestions system.

> Other examples of graphs include the result of a web crawl for a website or for the entire world wide web, city routes as seen on Google Maps, etc. Furthermore, different search engines have different web network models.

So this was the introductory lecture on Graph Theory. We'll be seeing modelling these graphs and their different algorithms in the upcoming lectures. You just stay tuned.

Thank you for being with me throughout the session. I hope you enjoyed it. If you appreciate my work, please let your friends know about this channel too. Lectures on graphs have just started, and if you haven't saved this already, you just have to move on to codewithharry.com or my YouTube channel to access them. See you all there in the next lecture. Till then keep learning.