**Graph**

Graph – set of values that are related in a pair-wise manner

Chart, radar chart

Description automatically generated

Consist of nodes that are connected by edges.

**Can be used in:**

* Friendships
* Family trees
* Networks on World Wide Web
* Roads and the roads connecting to it
* Facebook uses graph for social media
* Google Maps – used for determining the shortest path for the trip

**Advantages of Graph:**

* Relationships

**Disadvantages of Graph:**

* Scaling is hard

**Directed Graph**

Diagram

Description automatically generated

For example: This is like a one-way street. One direction only.

**Undirected Graph**

Diagram

Description automatically generated

For example: this is like a highway between 2 cities where you can go back and forth.

**Weighted Graph**

Diagram

Description automatically generated

There are values between nodes.

This is used by Google Maps to determine what is the shortest path to get from point A to point B.

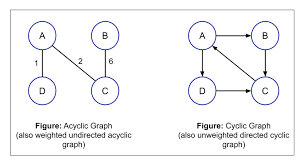
Used a lot in calculating optimal paths.

**Unweighted Graph**

Diagram, engineering drawing

Description automatically generated

**Cyclic Graph**

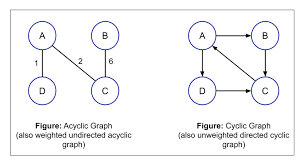
v

Vertices are connected in a circular fashion.

You can go back to the original node.

Common in weighted graphs such as Google Maps.

**Acyclic Graph**



You are not able to go back to the original node.

**Graphs Exercise**

<https://visualgo.net/en/graphds>

**Types of Graphs**

**Undirected Unweighted Cyclic Graph**

Chart

Description automatically generated with low confidence

**Undirected Weighted Cyclic Graph**

Diagram

Description automatically generated

It has values/weights on the edges making it weighted.

You can get to the original node making it cyclic.

It is undirected meaning you can go back and forth between each node.

**Directed Unweighted Acyclic Graph**

Diagram

Description automatically generated

Unable to go back to nodes making it acyclic.

Edges do not have values/weights making it unweighted.

Each node points to another node making it directed.

**Directed Weighted Acyclic Graph**

Chart, diagram

Description automatically generated

============================================================================================

**Methods of Representing Graphs**

We can represent the image in three different methods of graphs.

* Edge List
* Adjacent List
* Adjacent Matrix



// Edge List - shows the connection between nodes

const graph = [[0, 2], [2, 3], [2, 1], [1, 3]];

0 -> 2

2 -> 1

2 -> 3

1 -> 3

// Adjacent List - create a graph where the index is the node and the values are the nodes' neighbors

// We can use arrays, objects, or linked list

const graph = [[2], [2, 3], [0, 1, 3], [1, 2]];

node = 0

index[0] which is node 0 is connected to 2 (here we have [2] as the first value)

Index[1] which is node 1 is connected to [2, 3]

Index[2] which is node 2 is connected to [0, 1, 3]

Index[3] which is node 3 is connected to [1, 2]

// Adjacent Matrix 0's and 1's used to represent whether node x is connected to node y.

const graph = [

    [0, 0, 1, 0], // node[0] has connection with 2

    [0, 0, 1, 1], // node[1] has connection with 2 and 3

    [1, 1, 0, 1], // node[2] has connection with 0, 1, and 3

    [0, 1, 1, 0]  // node[3] has connection with 1 and 2

]