Graph Theory Overview

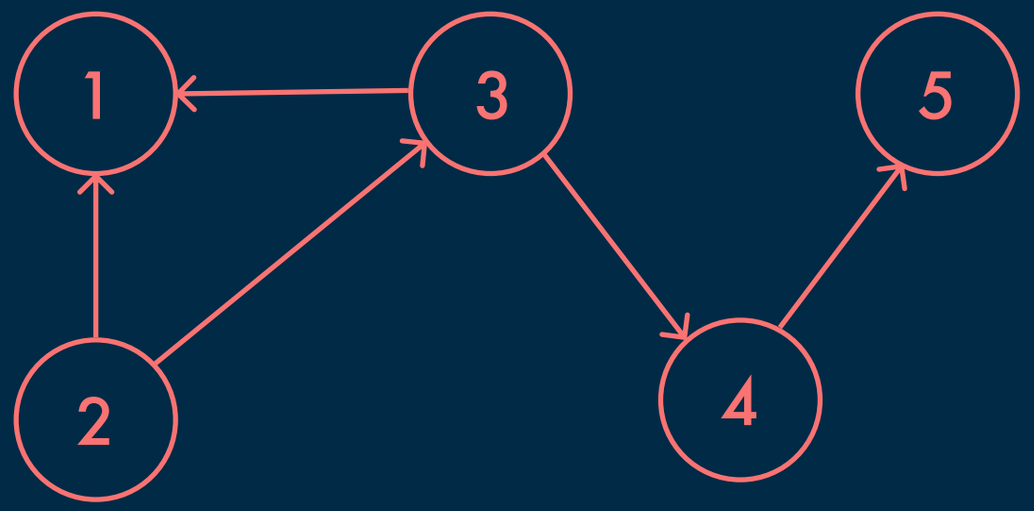
Kuanysheva Adema, Orynbayeva Leila, Azhar Ilyasova,
Tilek Akanbekova, Nurmadiyar Kydyraliyev

Contents

Graph Theory

In mathematics, graph theory is the study of graphs, being mathematical structures used to model paired interrelations between objects.

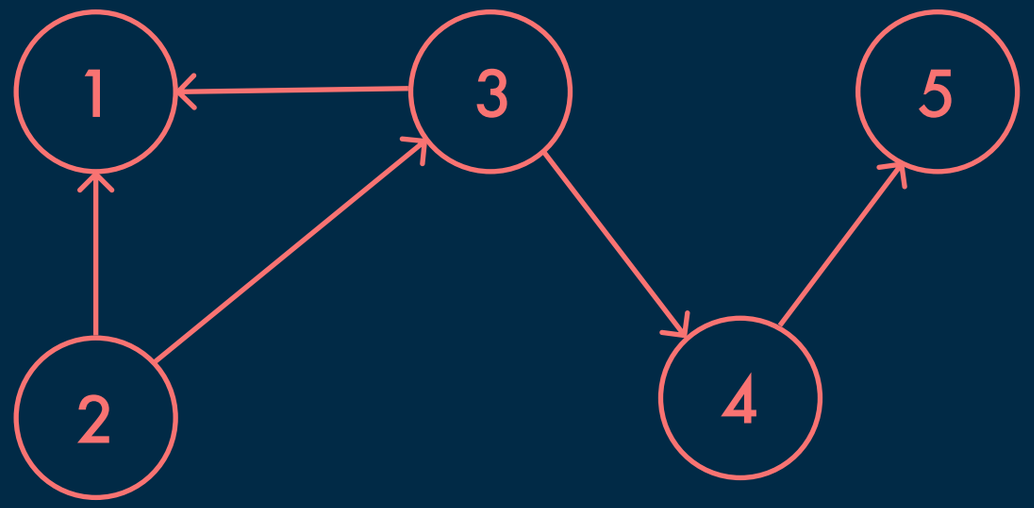
In terms of this context, a graph is composed of vertices (also called nodes or points) that are joined by edges (also called links or lines).



Directed Graph

A distinct type of graph, where edges link a pair of vertices asymmetrically.

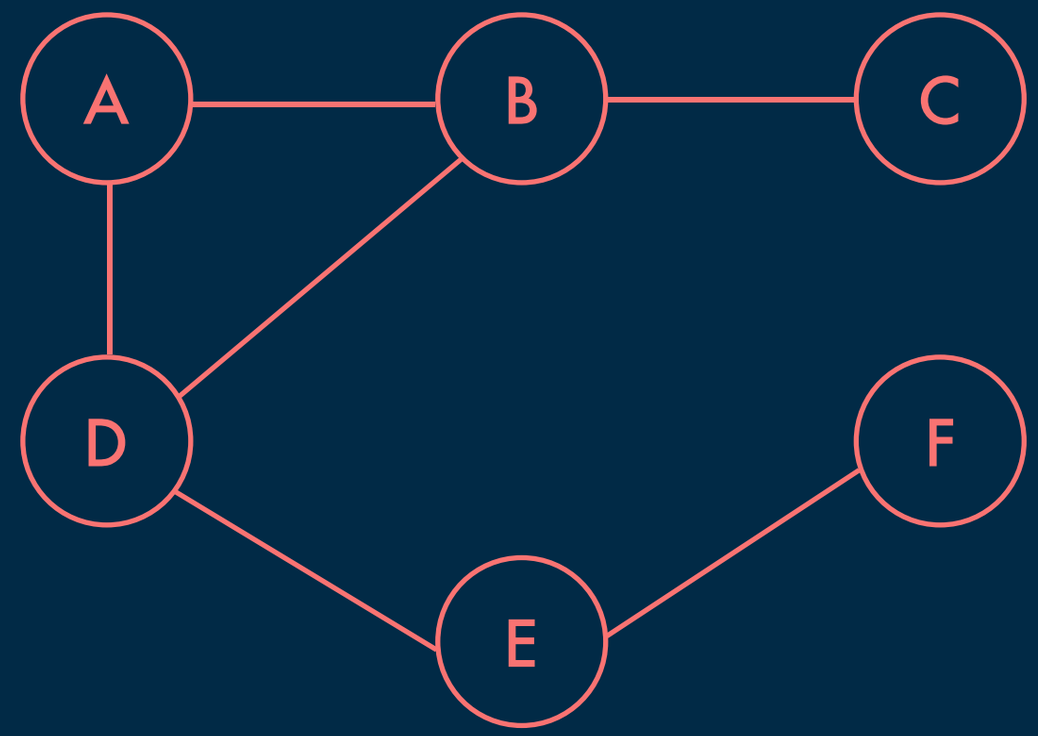
A Directed Graph G(V,E) is considered as a set of edges and vertices, where each edge has a direction.



Undirected Graph

A distinct type of graph, where edges link a pair of vertices symmetrically.

An Undirected Graph G(V,E) is considered as a set of edges and vertices, where each edge is bidirectional.



*edge (U, V) is equivalent to the edge (V,U)

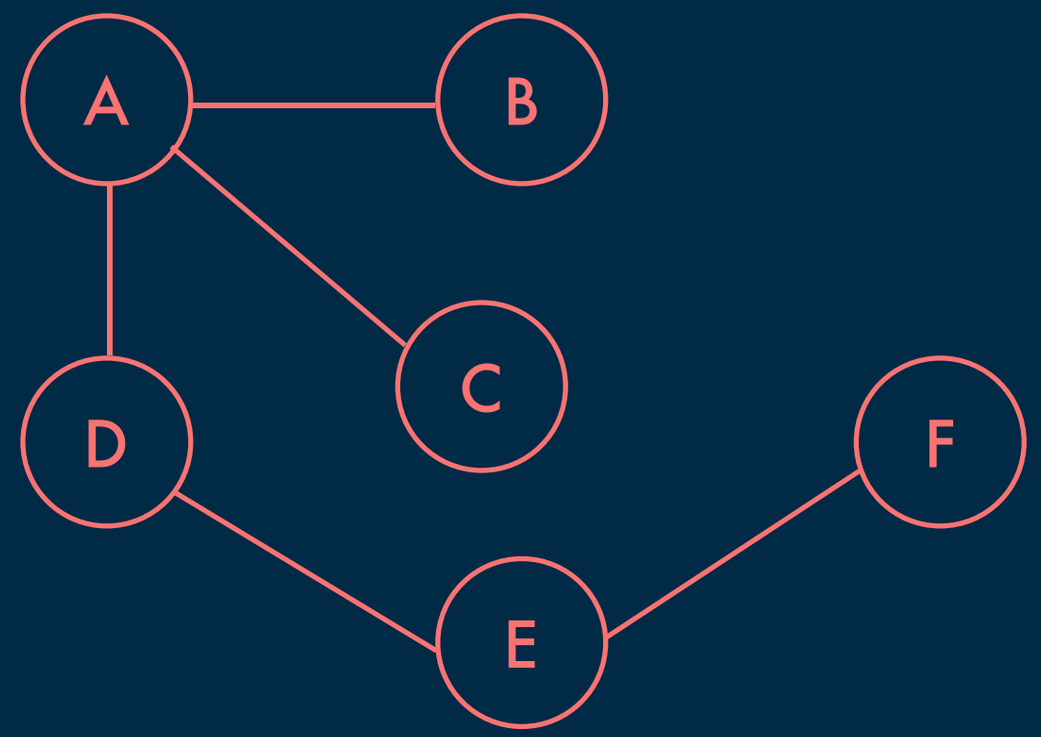
Graph Types

Trees

An undirected graph where two vertices form a pairwise relationship by exactly one path.

A graph is considered a tree if:

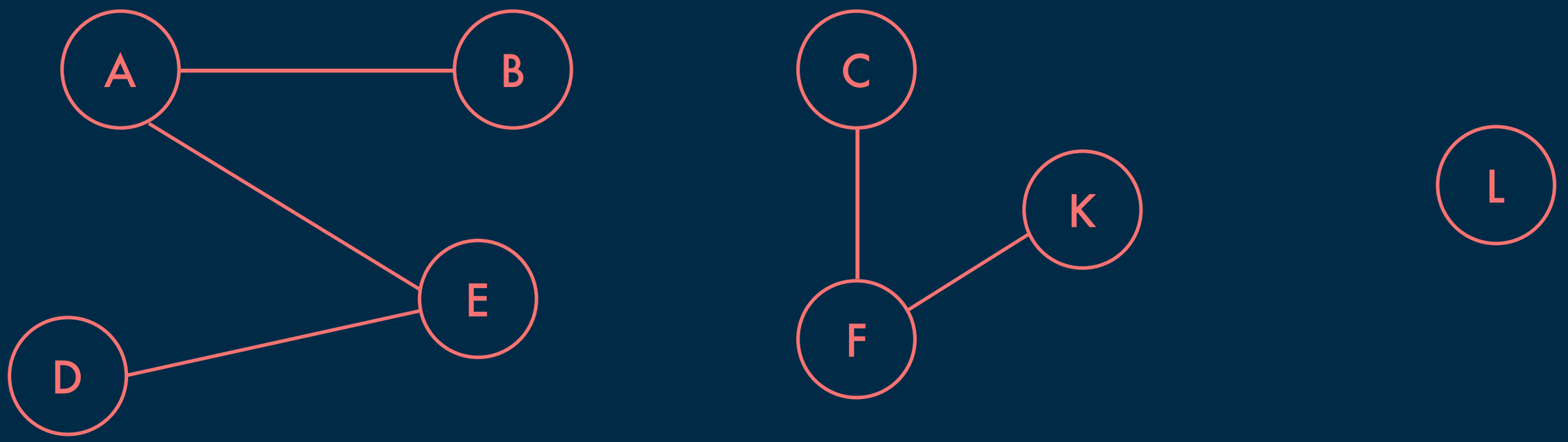
1. It is connected & contains no cycles(acyclic)
2. Has no simple cycles and has n-1 edges
3. Any two vertices can be merely connected by a unique simple path



Forest

An undirected graph where each two vertices are linked by at most one path. Forest is equivalent to:

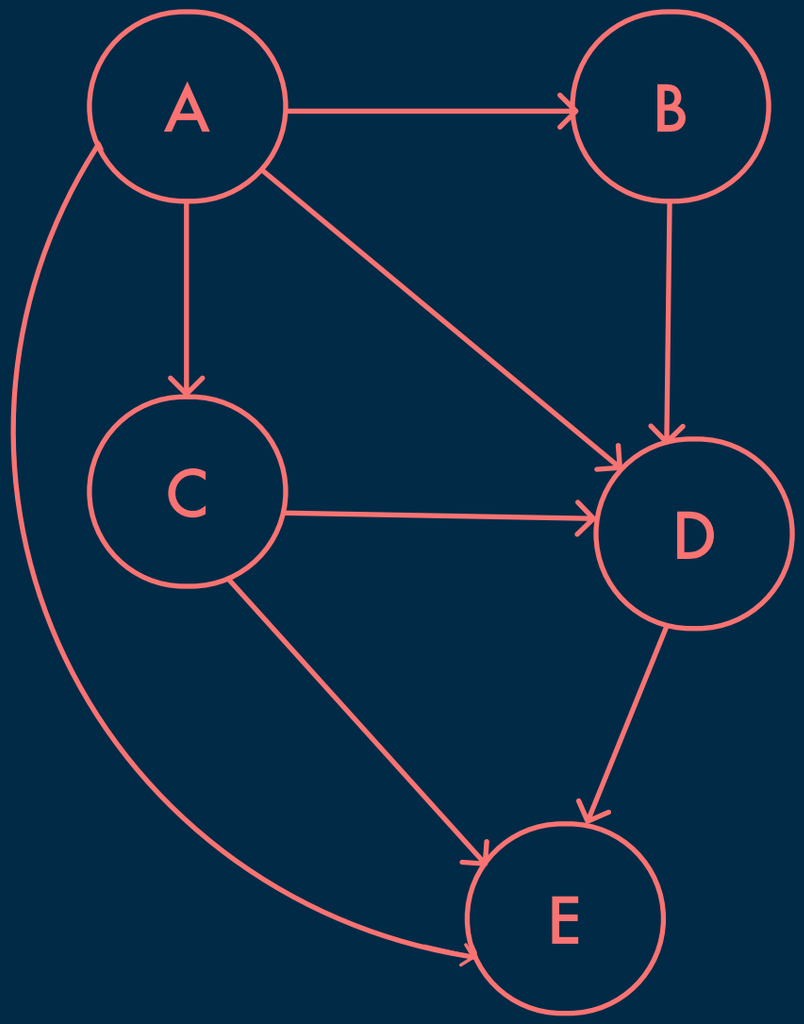
1. Acyclic undirected graph
2. Disjoint union of trees.



Directed Acyclic Graph

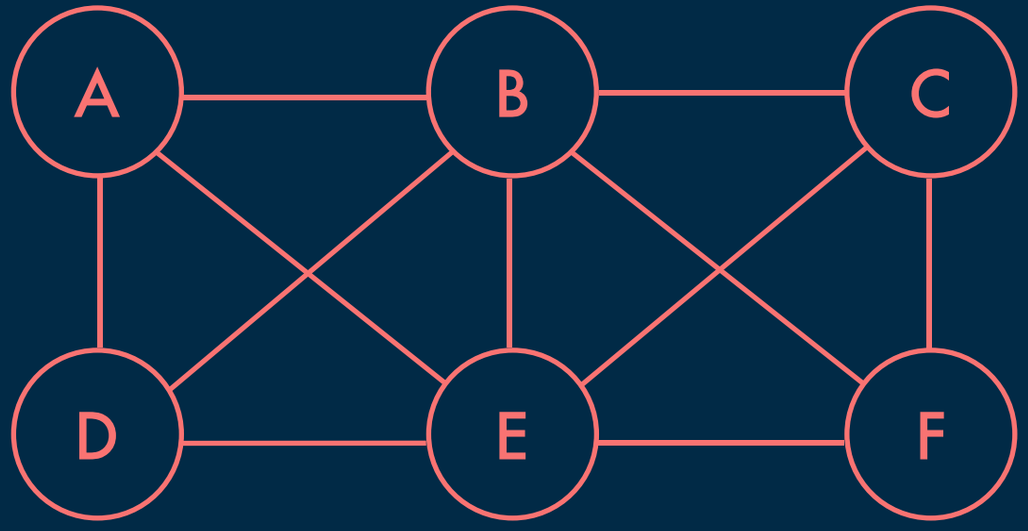
A directed graph with no directed cycles.

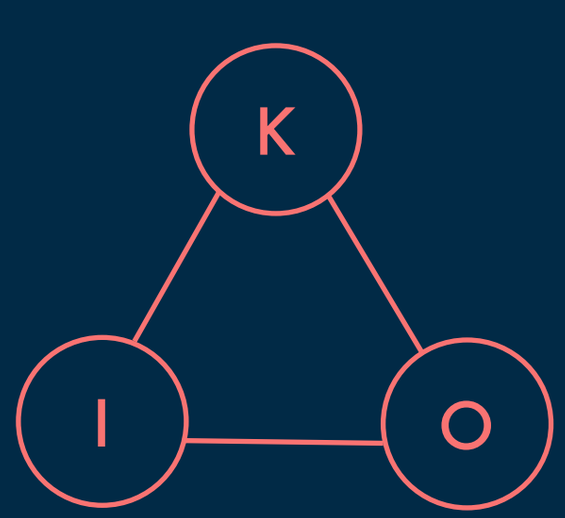
DAG consists of vertices and edges (also called arcs), where each edge that is directed from one vertex to another will never form a connection that creates a closed loop.



Complete Graph

An undirected graph where every pair of distinct vertices is connected by a pair of unique edges (one in each direction).





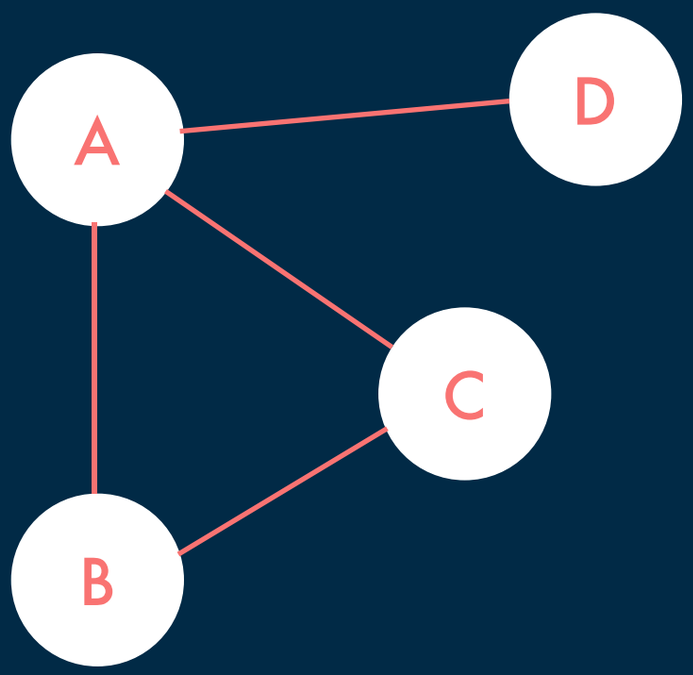
Representation of Graphs

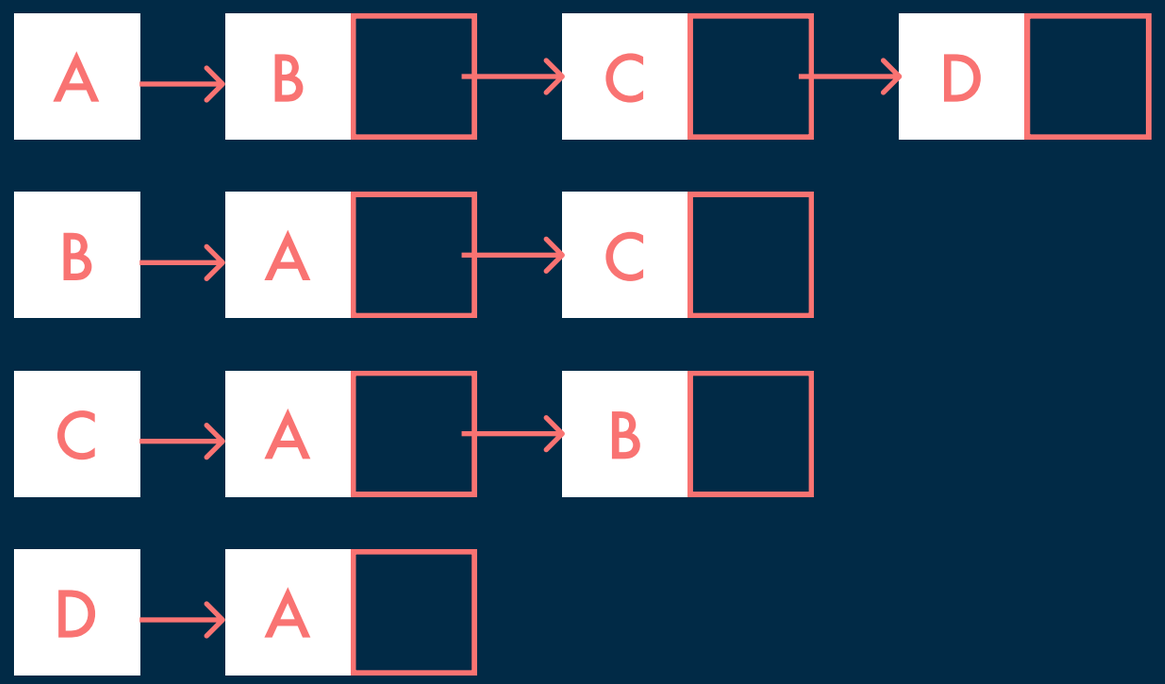
Using Programming Language

Adjacency List

A representation of a graph as an array of linked lists.

The index of the array represents a vertex and each element in its linked list represents the other vertices that form an edge with the vertex.





Adjacency List

Pros +

An adjacency list is efficient in terms of storage because we only need to store the values for the edges. For a sparse graph with millions of vertices and edges, this can mean a lot of saved space.

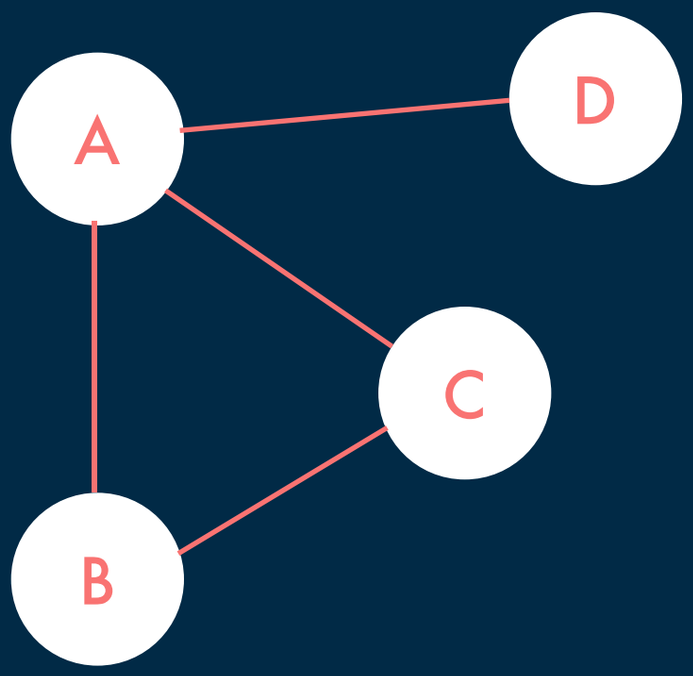
Cons -

An adjacency list is not efficient for edge weight lookup. It is slow – O(E), as we need to find it in linear time.

Adjacency Matrix

A a way of representing a graph G = {V, E} as a matrix of booleans.

The size of the matrix is VxV where V is the number of vertices in the graph and the value of an entry Aij is either 1 or 0 depending on whether there is an edge from vertex i to vertex j.



	A	B	C	D
A	0	1	1	1
B	1	0	1	C
C	1	1	0	0
D	1	0	0	0

Adjacency Matrix

Pros +

The basic operations like adding an edge, removing an edge and checking whether there is an edge from vertex i to vertex j are extremely time efficient, constant time operations.

Cons -

The VxV space requirement of the adjacency matrix makes it a memory hog.

Applications

Graph Theory Applications

1. Finding communities in networks, such as social media (friend/ connection recommendations).
2. Ranking and ordering hyperlinks in search engines.
3. Google Maps/GPS to find shortest path to the destination.
4. DNA sequencing.
5. Computer network security.

and so on...

Thank You !