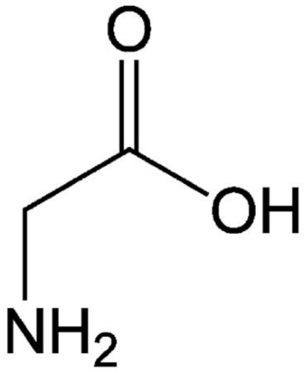
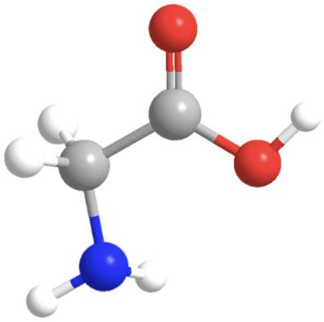
Graph Neural Networks

GNN

Examples	Nodes	Edges	Example Usages
Google Knowledge Graph	People, Places, Things	Connections	SEO
Chemical Molecular Structure	Atoms	Bonds	Molecule Structure
Document citation Network	Documents	Citation by a person	Cora Dataset
Social Media Networks	Person, properties	Connections	Virality, Influence
Network Design Security	Devices	Connections	Relationships
Financial Transactions	Transections	Connectivity	Fraud, AML

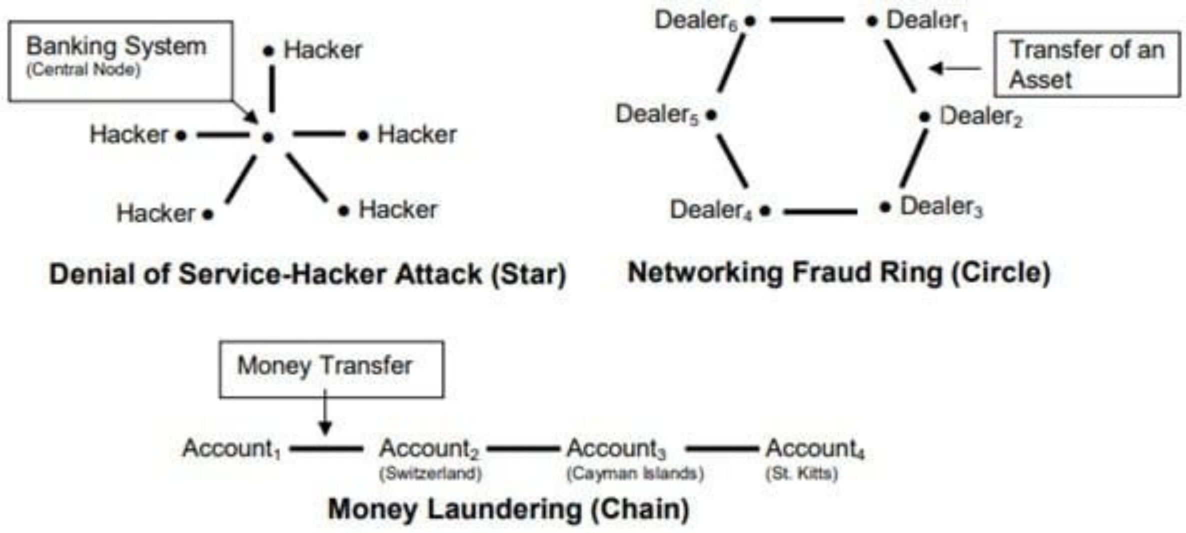
Example	Nodes	Edges	Example
Chemical Molecular Structure	Atoms	Bonds	Molecule Structure (Glycine)





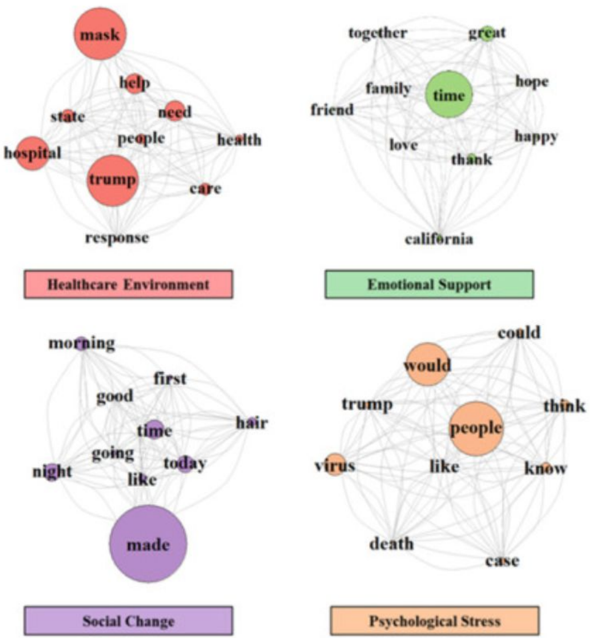
https://www.acs.org/content/acs/en/molecule-of-the-week/archive/g/glycine.html

Example	Nodes	Edges	Example Usages
Financial Transactions	Transections	Connectivity	Fraud, AML



https://www.cgi.com/sites/default/files/white-papers/Implementing-social-network-analysis-for-fraud-prevention.pdf

Example	Nodes	Edges	Example Usages
Social Media Networks	Person, Entities	Connections	Virality, Influence

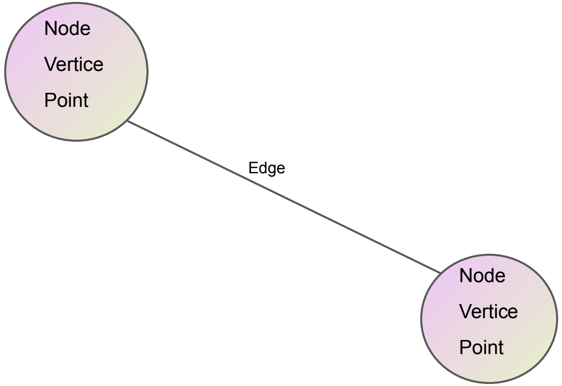


Social network centrality measures of the top 10 words on major COVID-19 themes.

Themes	Degree	Betweenness	Closeness	Eigenvector
Healthcare Environment	18	4.0	0.001885	0.5443
Emotional Support	18	3.6	0.009339	0.5834
Business Economy	18	1.3	0.000421	0.6495
Social Change	18	5.0	0.002602	0.5315
Psychological Stress	18	2.2	0.000656	0.5790

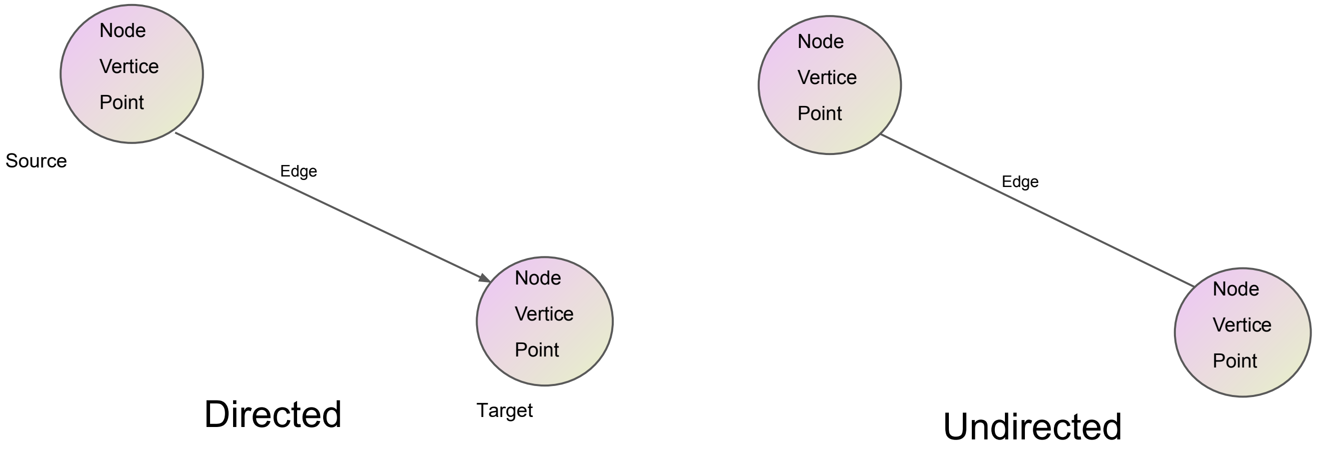
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7438102/

Fundamentals of Graphs



A Graph is a collection of vertices and edges.
Node, Vertice, Point:

You Define what part of data will be used as Node, vertice or Point, It’s your design.



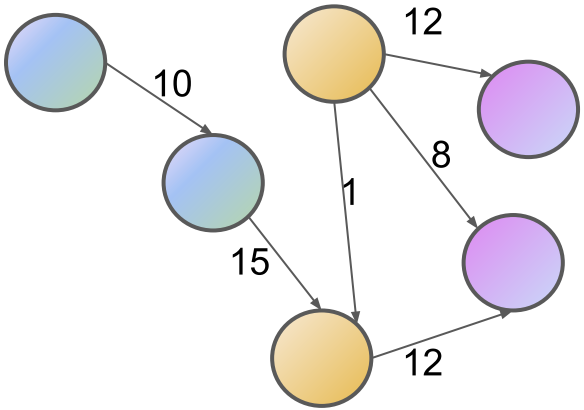
Node, Vertice, Point:

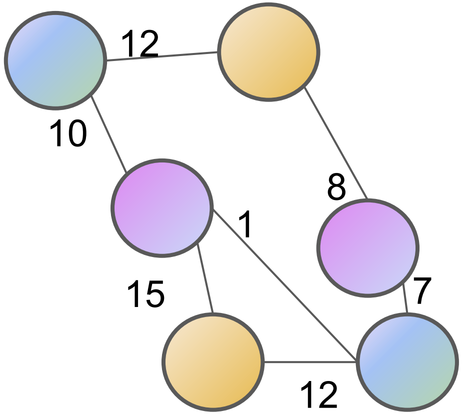
You Define what part of data will be used as Node, vertice or Point, It’s your design.

Directed Vs Undirected Graph

https://docs.google.com/presentation/d/11mR1nkIR9fbHegFkcFq8z9oDQ5sjv8E3JJp1LfLGKuk/edit#slide=id.g78327f1586_0_1154

Weighted Graphs





- A Weighted graph is a graph with edges labeled by the numbers
    - I.e. Distance, quantity, price, value etc.
- A weight is a numerical value attached to each individual edge.
- Each branch must have some weight as defined in the weight rule

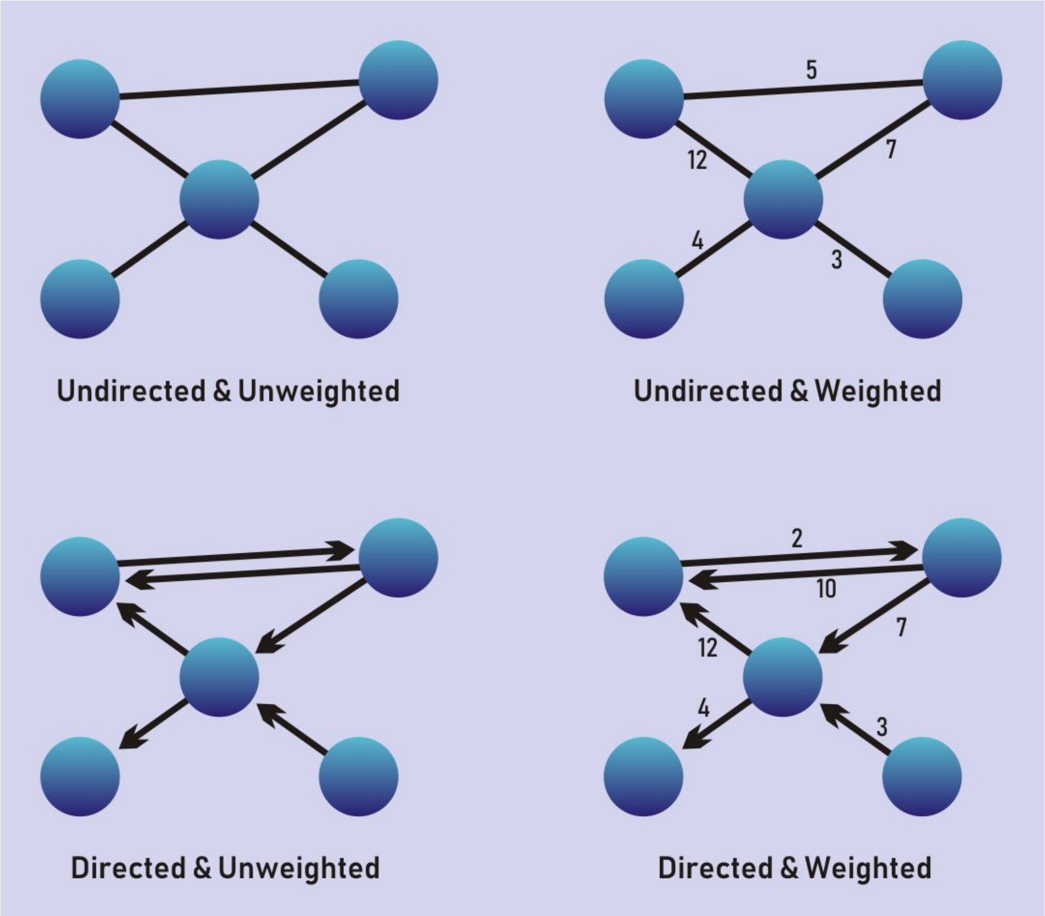


Image Source: https://medium.com/tebs-lab/types-of-graphs-7f3891303ea8

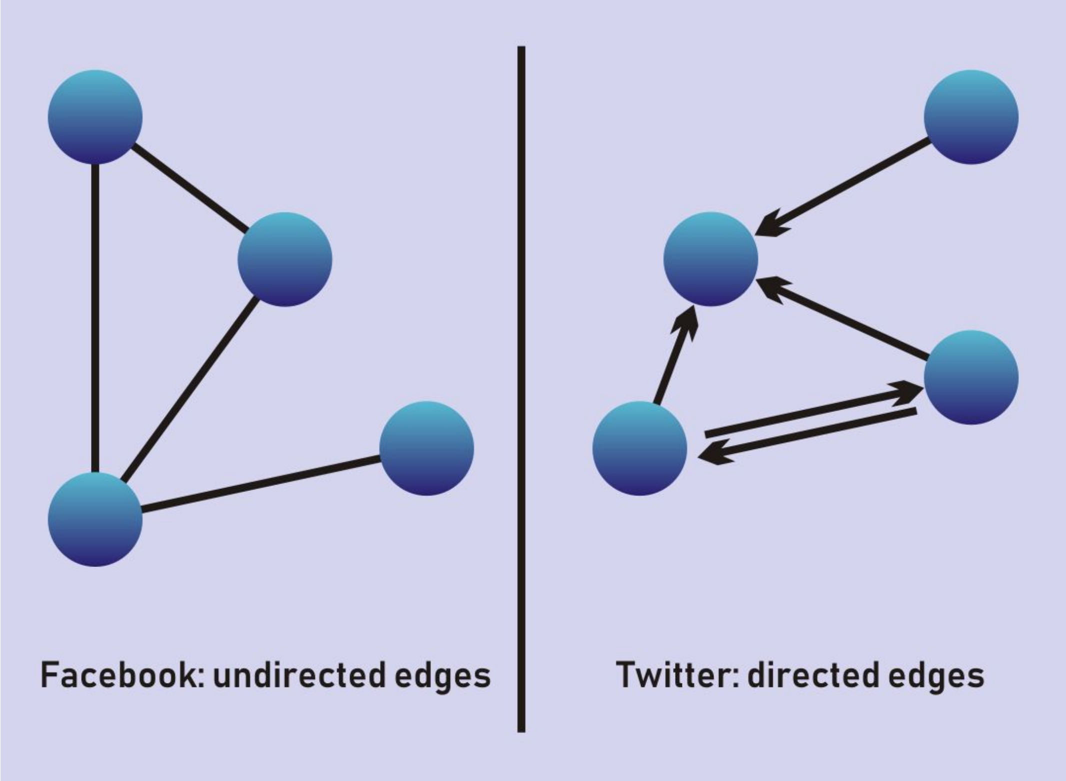
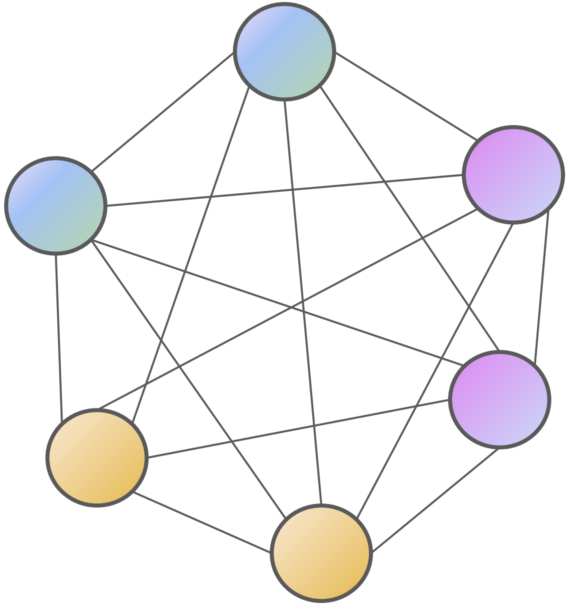


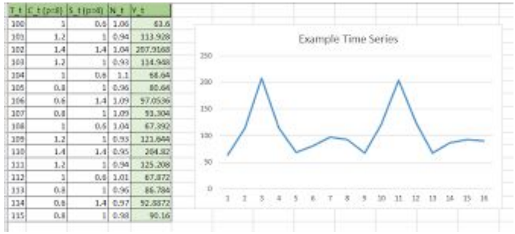
Image Source: https://medium.com/tebs-lab/types-of-graphs-7f3891303ea8

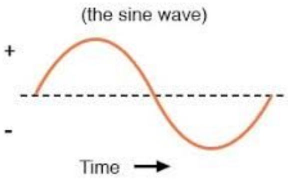
Complete Graph
Fully Connected Graph

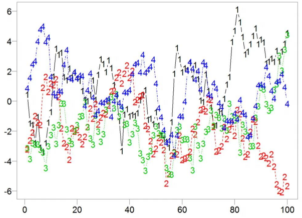


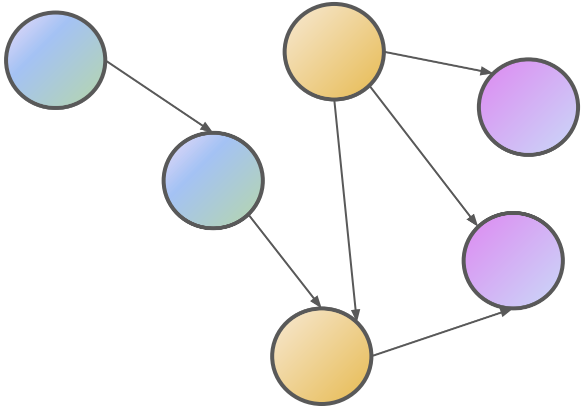
All nodes are connected with each other

Why Graphs are hard to understand?





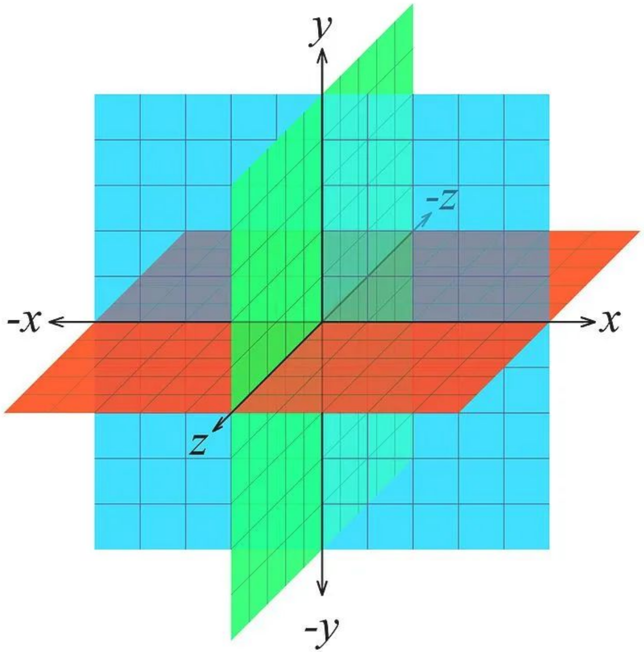


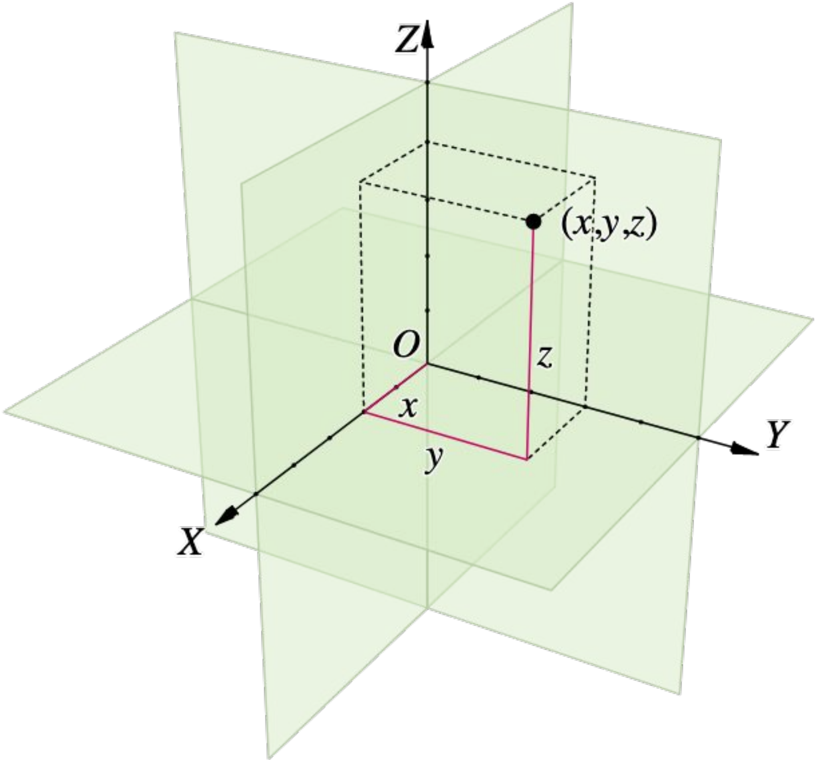


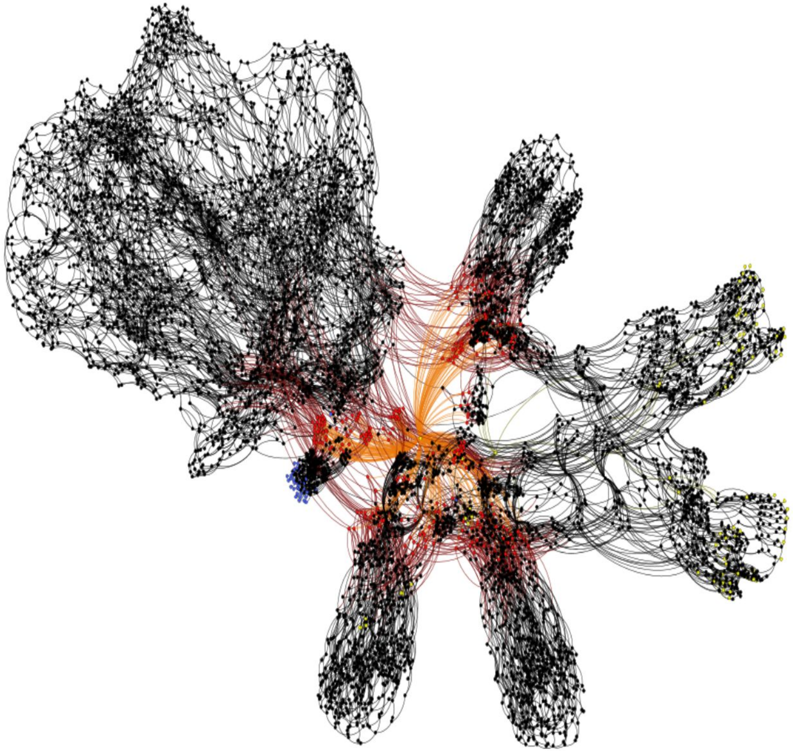
Can be represented into the Euclidean space also have the fixed form or representation.

Graphs does not have a fixed form and can NOTE be represented into the Euclidean space

Euclidean Space - 3 Dimensional (x,y,z) Plane





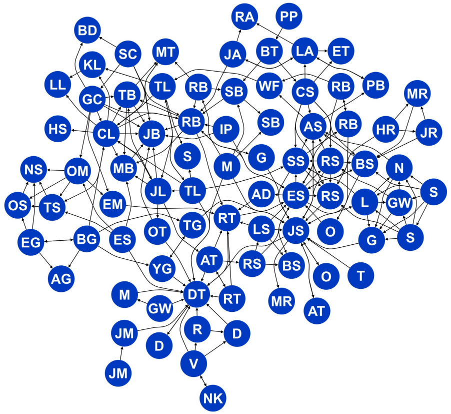


Example of a giant graph: circuit netlist. Figure from J. Baehr et. al. “Machine Learning and Structural Characteristics of Reverse Engineering”

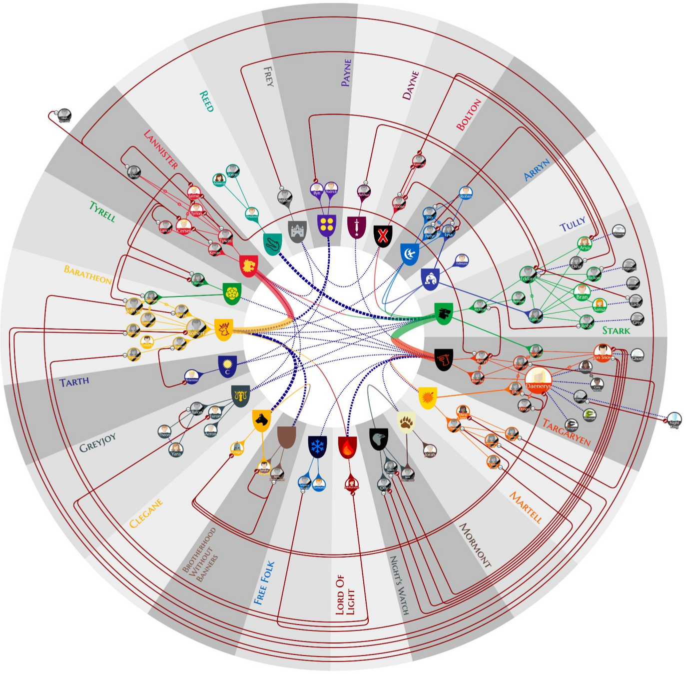
Why we should use Graphs?

- Give intuitive representation to abstract concepts i.e. relationships and interactions.
- Intuitively visual representation of information.
- Form a Natural basis for analyzing relationships in a Social context.
- Break down complex problems into simpler representations
- Transform the complex problems into representations from different perspectives.

‘Game of Thrones’ Relationship Graph

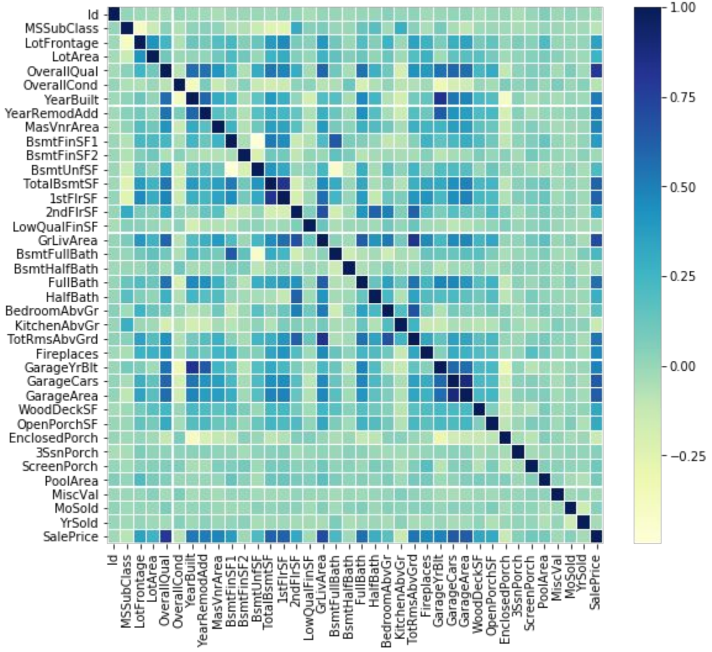


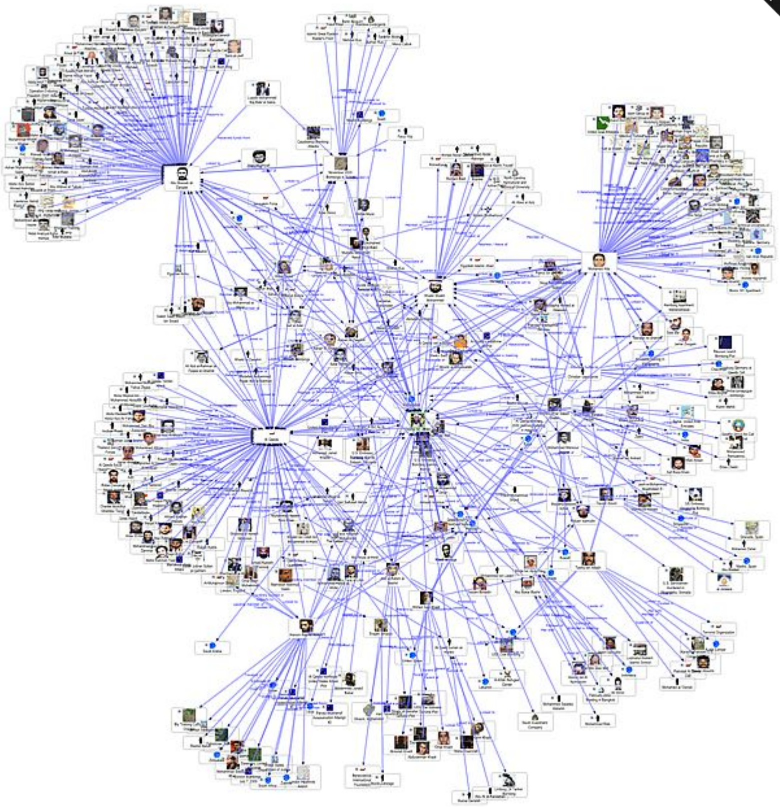
84 Characters

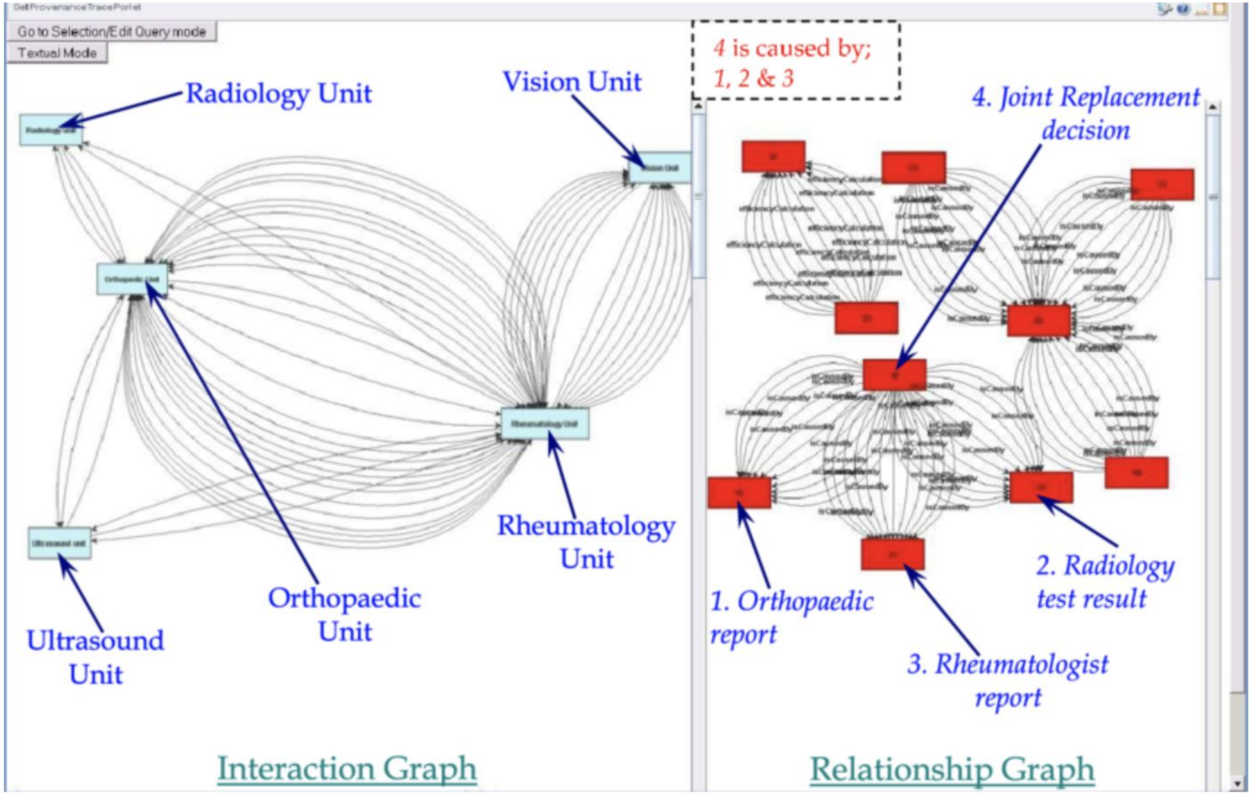


https://www.yworks.com/blog/graph-drawing-contest-2018

Relation & Correlation







https://www.researchgate.net/figure/Interaction-graph-and-relationship-graph_fig1_221158679

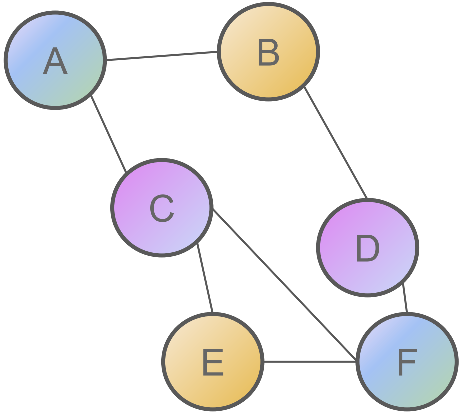
Traditional Graph Analysis Methods

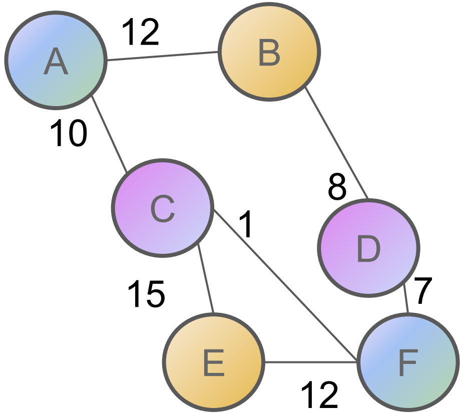
- Searching algorithms, e.g. BFS, DFS
- Shortest path algorithms, e.g. Dijkstra’s algorithm, Nearest Neighbour
- Spanning-tree algorithms, e.g. Prim’s algorithm
- Clustering methods, e.g. Highly Connected Components, k-mean

Limited based on their use cases

Mathematics of Graph

Mathematical Representation of Graph





Set of Vertices

- V = {A, B, C, D, E, F} ->

Set of Edges

- E = {AB, AC, BD, CE, CF, DF, EF}
- E = {(A,B), (A,C), (B,D), (C,E), (C,F), (D,F), (E,F)}

Set of Vertices

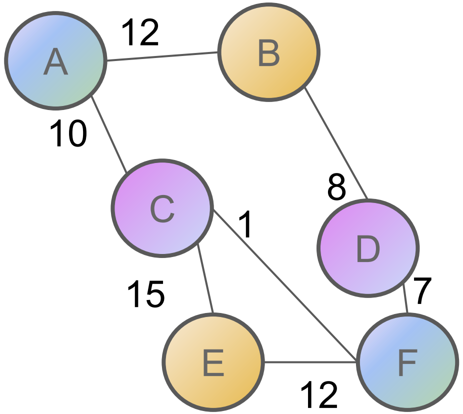
- V = {A, B, C, D, E, F} ->

Set of Edges

- E = {(A,B,12), (A,C,10), (B,D,8), (C,E,15), (C,F,1), (D,F,7), (E,F,12)}

Graph G = (V, E)

Neighbors



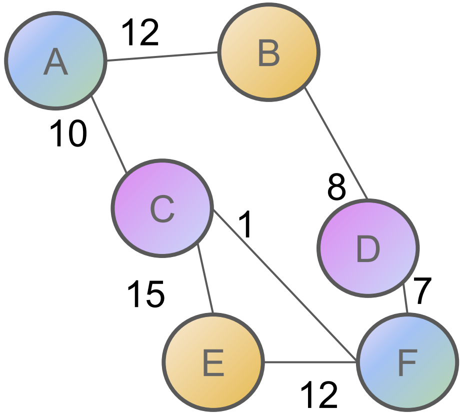
Neighbors:

- Two nodes that are connected with an edge are called neighbors.

Given Example:

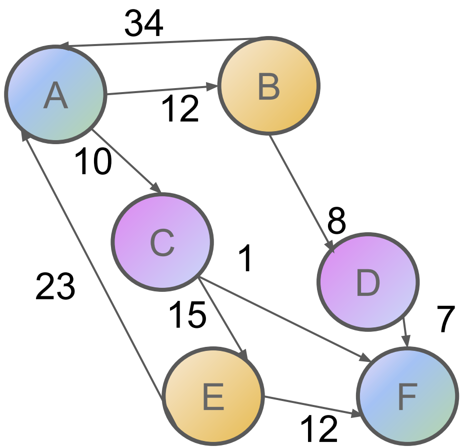
- B and C are neighbors for A
- E and F are neighbors for C
- D is neighbor for B
- F is neighbor for C, D and E

Edge List



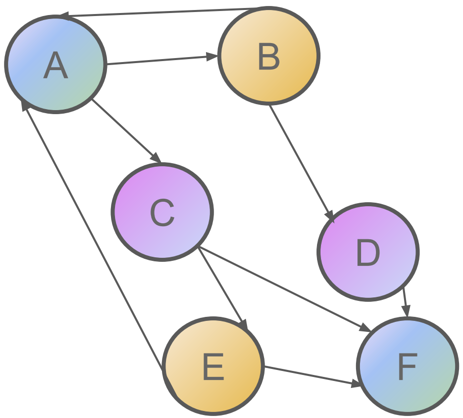
A	B	12
A	C	10
B	D	8
C	E	15
C	F	1
D	F	7
E	F	12

Edge List - Directed



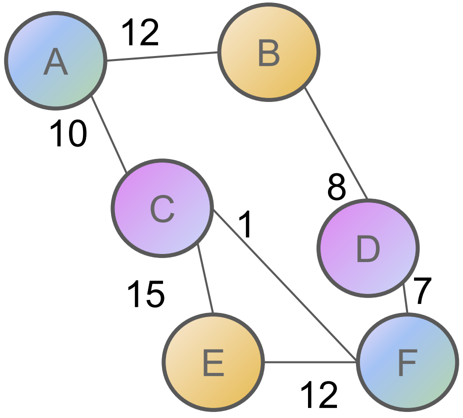
A	B	12
A	C	10
B	D	8
C	E	15
C	F	1
D	F	7
E	F	12
E	A	23
B	A	34

Edge List - Directed & Un-weighted



A	B
A	C
B	D
C	E
C	F
D	F
E	F
E	A
B	A

Adjacency Matrix

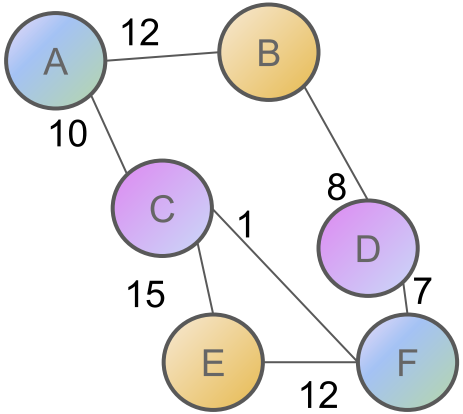


- Adjacency matrix a 2D square matrix
- Each node in the graph has an entry in both dimensions.
- Unweighted graph as T/F or 1/0 values
- Weighted graph as weights, no weights means -1

Representation:

- $A = N \, x \, N$

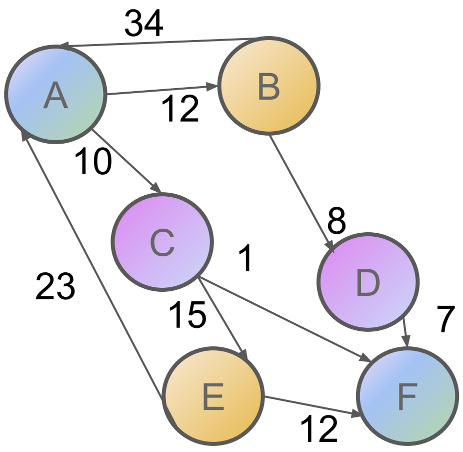
Adjacency Matrix - Weighted & Undirected



	A	B	C	D	E	F
A	-1	12	10	-1	-1	-1
B	-1	-1	-1	8	-1	-1
C	-1	-1	-1	-1	15	1
D	-1	-1	-1	-1	-1	7
E	-1	-1	-1	-1	-1	12
F	-1	-1	-1	-1	-1	-1

$A = 6 \: x \: 6$

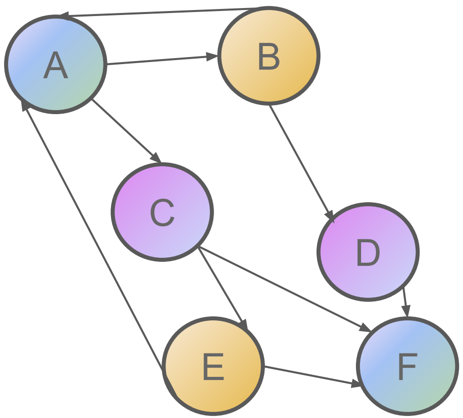
Adjacency Matrix - Weighted & Directed



	A	B	C	D	E	F
A	-1	12	10	-1	-1	-1
B	34	-1	-1	8	-1	-1
C	-1	-1	-1	-1	15	1
D	-1	-1	-1	-1	-1	7
E	23	-1	-1	-1	-1	12
F	-1	-1	-1	-1	-1	-1

$A = 6 \, x \, 6$

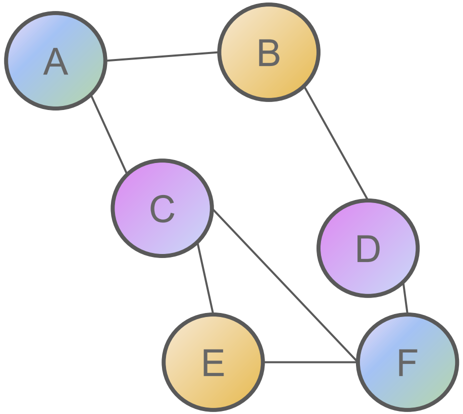
Adjacency Matrix - Un-Weighted & Directed



	A	B	C	D	E	F
A	F	T	T	F	F	F
B	T	F	F	T	F	F
C	F	F	F	F	T	T
D	F	F	F	F	F	T
E	T	F	F	F	F	T
F	F	F	F	F	F	F

$A = 6 \, x \, 6$

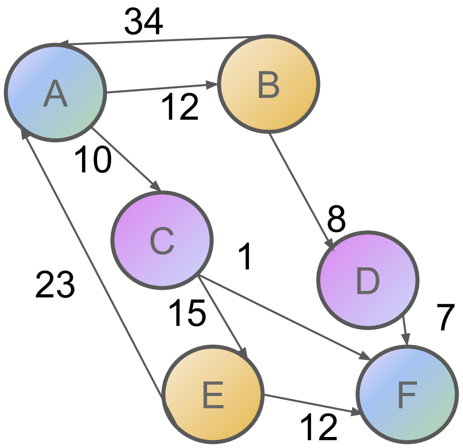
Adjacency Matrix - Unweighted & Undirected



	A	B	C	D	E	F
A	F	T	T	F	F	F
B	F	F	F	T	F	F
C	F	F	F	F	T	T
D	F	F	F	F	F	T
E	F	F	F	F	F	T
F	F	F	F	F	F	F

$A = 6 \, x \, 6$

Adjacency Matrix - Weighted & Directed



	A	B	C	D	E	F
A	-1	12	10	-1	-1	-1
B	34	-1	-1	8	-1	-1
C	-1	-1	-1	-1	15	1
D	-1	-1	-1	-1	-1	7
E	23	-1	-1	-1	-1	12
F	-1	-1	-1	-1	-1	-1

A_{AB}	A-B
A_{AC}	A-C
A_{AF}	A-C-E-F
A_{CD}	C-E-A-B-D
A_{BC}	X

$$A = 6 \, x \, 6$$

Complete Graph

- All elements of adjacency matrix **A** are 1/T, except along the diagonal
- Path exists for each and every node

Sparse Graph

- Not all elements of adjacency matrix **A** are 1/T, lots of values are -1 or F/0
- Not every node is connected to each other

Extended Reading:

- https://medium.com/@TebbaVonMathenstien/implementations-of-graphs-92eb7f121793

Node Attribute Matrix / Feature Matrix (X)

- The features or attributes of each node
- A Graph with N nodes with the size of node attributes as F,
  - Matrix Shape = $N \times F$

Example:

Document 1: I travelled to Himalaya.
Document 2: I travelled to Las Vegas Nevada

Corpus: {I, travelled, to, Himalaya, Las, Vegas, Nevada}
Size of Corpus (F) = 7

	Document 1	**Document 2**
I	1	1
Travelled	1	1
to	1	1
Himalaya	1	0
Las	0	1
Vegas	0	1
Nevada	0	1

The shape of node attributes matrix X = NxF = 2x7 = 14

Bag-of-words Illustration as node features

Trick Question

All Diagonals are values are 1

	A	B	C	D	E	F
A	T	T	T	F	F	F
B	F	T	F	T	F	F
C	F	F	T	F	T	T
D	F	F	F	T	F	T
E	F	F	F	F	T	T
F	F	F	F	F	F	T

Trick Question - **Self Loop**

All Diagonals are values are 1

	A	B	C	D	E	F
A	T	T	T	F	F	F
B	F	T	F	T	F	F
C	F	F	T	F	T	T
D	F	F	F	T	F	T
E	F	F	F	F	T	T
F	F	F	F	F	F	T

Graph Representation

Graph Representation

Adjacency Matrix (A)

- Square Matrix (A = NxN)

Incidence (Relation Matrix) Matrix (I)

- Nodes n with edges m will have (I = nxm)

Degree Matrix (D)

- Diagonal Matrix (D)

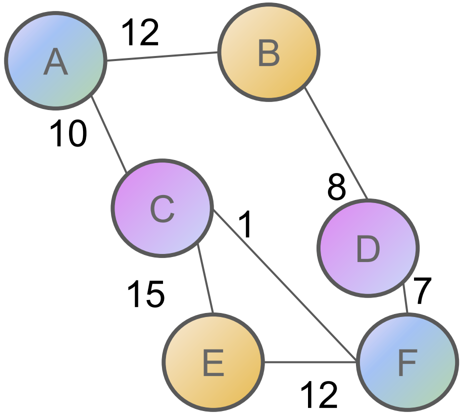
Laplacian Matrix:

- L = D - A

Bag of Nodes

- Aggregate node level features using some mathematical approach i.e. taking the mean of the node degrees or histogram of the edge connections

Adjacency Matrix

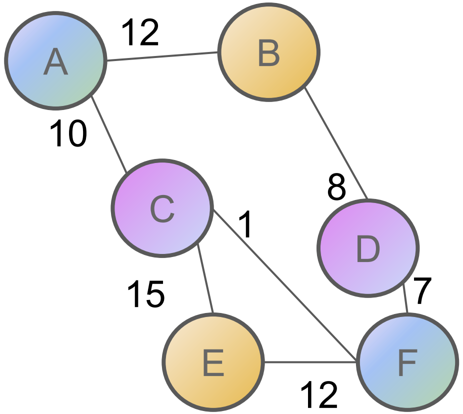


- Adjacency matrix a 2D square matrix
- Each node in the graph has an entry in both dimensions.
- Unweighted graph as T/F or 1/0 values (Binary Value Adjacency Matrix)
- Weighted graph as weights, no weights means -1 (Weighted Adjacency Matrix)

Representation:

- $A = N \, x \, N$

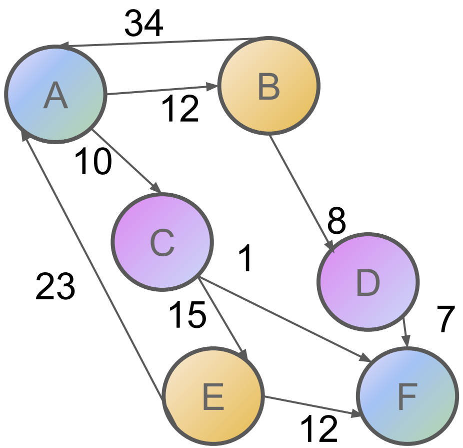
Adjacency Matrix - Weighted & Undirected



	A	B	C	D	E	F
A	-1	12	10	-1	-1	-1
B	-1	-1	-1	8	-1	-1
C	-1	-1	-1	-1	15	1
D	-1	-1	-1	-1	-1	7
E	-1	-1	-1	-1	-1	12
F	-1	-1	-1	-1	-1	-1

$$A = 6 \, x \, 6$$

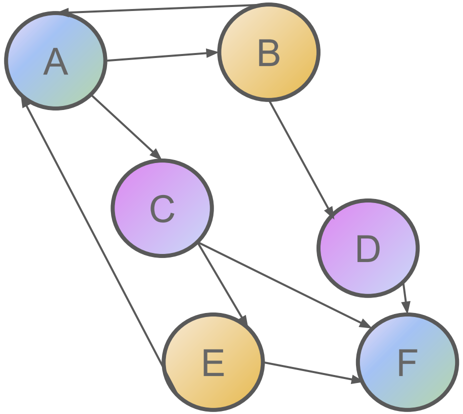
Adjacency Matrix - Weighted & Directed



	A	B	C	D	E	F
A	-1	12	10	-1	-1	-1
B	34	-1	-1	8	-1	-1
C	-1	-1	-1	-1	15	1
D	-1	-1	-1	-1	-1	7
E	23	-1	-1	-1	-1	12
F	-1	-1	-1	-1	-1	-1

$A = 6 \, x \, 6$

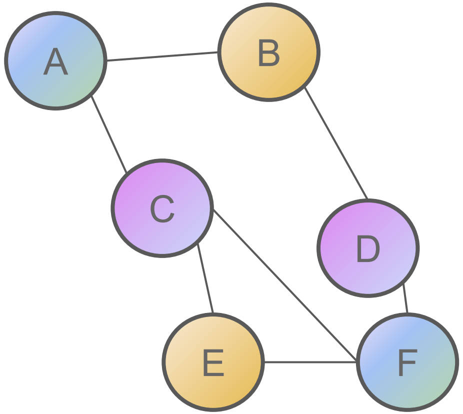
Adjacency Matrix - Un-Weighted & Directed



	A	B	C	D	E	F
A	F	**T**	**T**	F	F	F
B	**T**	F	F	**T**	F	F
C	F	F	F	F	**T**	**T**
D	F	F	F	F	F	**T**
E	**T**	F	F	F	F	**T**
F	F	F	F	F	F	F

$$A = 6 \, x \, 6$$

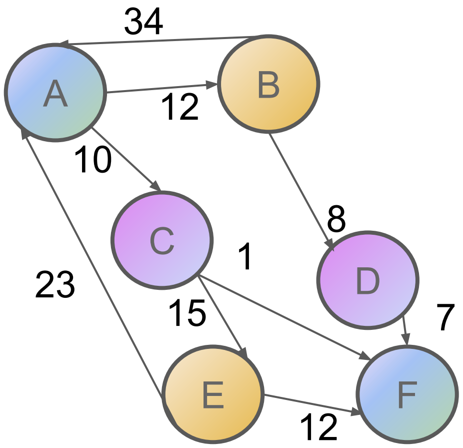
Adjacency Matrix - Unweighted & Undirected



	A	B	C	D	E	F
A	F	**T**	**T**	F	F	F
B	F	F	F	**T**	F	F
C	F	F	F	F	**T**	**T**
D	F	F	F	F	F	**T**
E	F	F	F	F	F	**T**
F	F	F	F	F	F	F

$$A = 6 \, x \, 6$$

Adjacency Matrix - Weighted & Directed

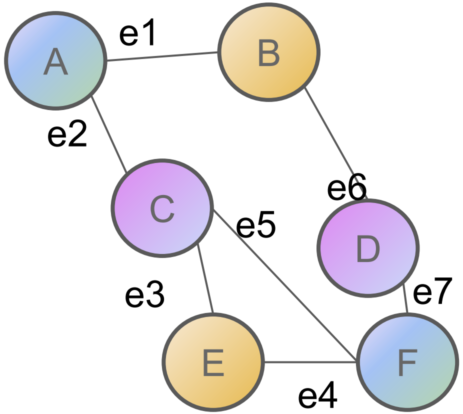


	A	B	C	D	E	F
A	-1	12	10	-1	-1	-1
B	34	-1	-1	8	-1	-1
C	-1	-1	-1	-1	15	1
D	-1	-1	-1	-1	-1	7
E	23	-1	-1	-1	-1	12
F	-1	-1	-1	-1	-1	-1

A_{AB}	A-B
A_{AC}	A-C
A_{AF}	A-C-E-F
A_{CD}	C-E-A-B-D
A_{BC}	X

$$A = 6 \, x \, 6$$

Incidence Matrix - Nodes N/Edge M - **Undirected** Graph

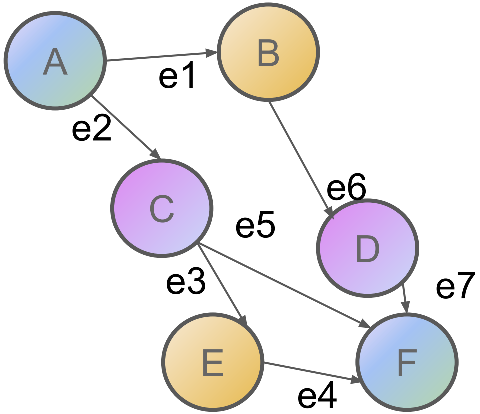


	e1	e2	e3	e4	e5	e6	e7
A	1	1	0	0	0	0	0
B	1	0	0	0	0	1	0
C	0	1	1	0	1	0	0
D	0	0	0	0	0	1	1
E	0	0	1	1	0	0	0
F	0	0	0	1	1	0	1

If Node and Edge are connected then value is 1 otherwise the value is 0.
Nodes are represented as **ROWS** and edges are as **COLUMNS**

$$I = 6 \; x \; 7$$

Incidence Matrix - Nodes N/Edge M - **Directed** Graph



$$I = 6 \times 7$$

	e1	e2	e3	e4	e5	e6	e7
A	-1	-1	0	0	0	0	0
B	1	0	0	0	0	-1	0
C	0	1	-1	0	-1	0	0
D	0	0	0	0	0	1	-1
E	0	0	1	-1	0	0	0
F	0	0	0	1	1	0	1

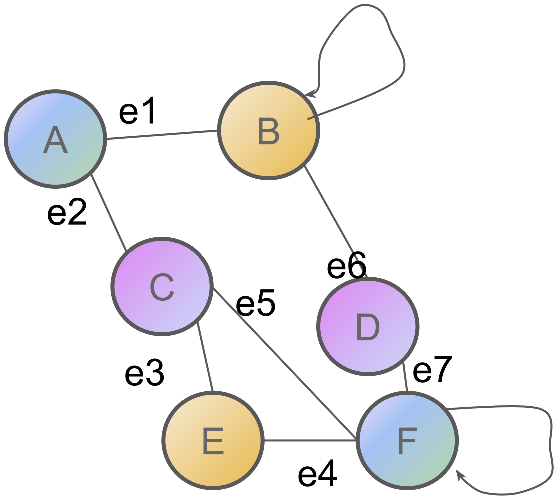
If Node and Edge are connected then value -1/1, otherwise the value is 0.

- If Node to **outward** direction connection then the value is -1
- If node has **inwards** direction connection then the value is 1

Nodes are represented as **ROWS** and edges are as **COLUMNS**

Degree Matrix - **Directed/Undirected** Graph (Same)

Diagonal Matrix



Degree = Number of edges connected to each node

	A	B	C	D	E	F
A	2	0	0	0	0	0
B	0	4	0	0	0	0
C	0	0	3	0	0	0
D	0	0	0	2	0	0
E	0	0	0	0	2	0
F	0	0	0	0	0	5

Diagonal

If Node has connection from edge then value is 1 otherwise the value is 0.
Nodes are represented as **ROWS** and **COLUMNS**

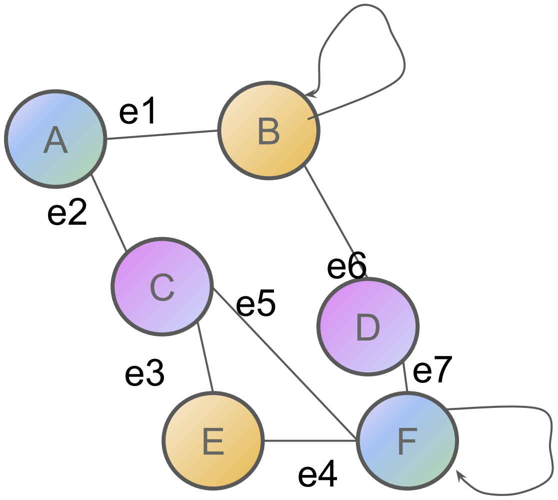
$$D = 6\ x\ 6$$

Laplacian Matrix - Graph Laplacian Matrix

- Measure of the smoothness of the matrix
- How quickly it changes between the Adjacent Vertices
- L = Diagonal Matrix - Adjacency Matrix
- L = Number of Edges connected to Node - Adjacency Matrix
- L = {D - A}

$$L = \{D - A\}$$

Laplacian Matrix - Graph Laplacian Matrix



	A	B	C	D	E	F
A	2	0	0	0	0	0
B	0	4	0	0	0	0
C	0	0	3	0	0	0
D	0	0	0	2	0	0
E	0	0	0	0	2	0
F	0	0	0	0	0	5

	A	B	C	D	E	F
A	0	1	1	0	0	0
B	1	2	0	1	0	0
C	1	0	0	0	1	1
D	0	1	0	0	0	1
E	0	0	1	0	0	1
F	0	0	1	1	1	2

$L = \{D - A\}$

D

A

Self Loop is considered as **2** nodes

Laplacian Matrix - Graph Laplacian Matrix

	A	B	C	D	E	F
A	2	0	0	0	0	0
B	0	4	0	0	0	0
C	0	0	3	0	0	0
D	0	0	0	2	0	0
E	0	0	0	0	2	0
F	0	0	0	0	0	5

D

-

	A	B	C	D	E	F
A	0	1	1	0	0	0
B	1	2	0	1	0	0
C	1	0	0	0	1	1
D	0	1	0	0	0	1
E	0	0	1	0	0	1
F	0	0	1	1	1	2

A

=

	A	B	C	D	E	F
A	2	-1	-1	0	0	0
B	-1	2	0	-1	0	0
C	-1	0	3	0	-1	-1
D	0	-1	0	2	0	-1
E	0	0	-1	0	2	-1
F	0	0	-1	-1	-1	3

$L = \{D - A\}$