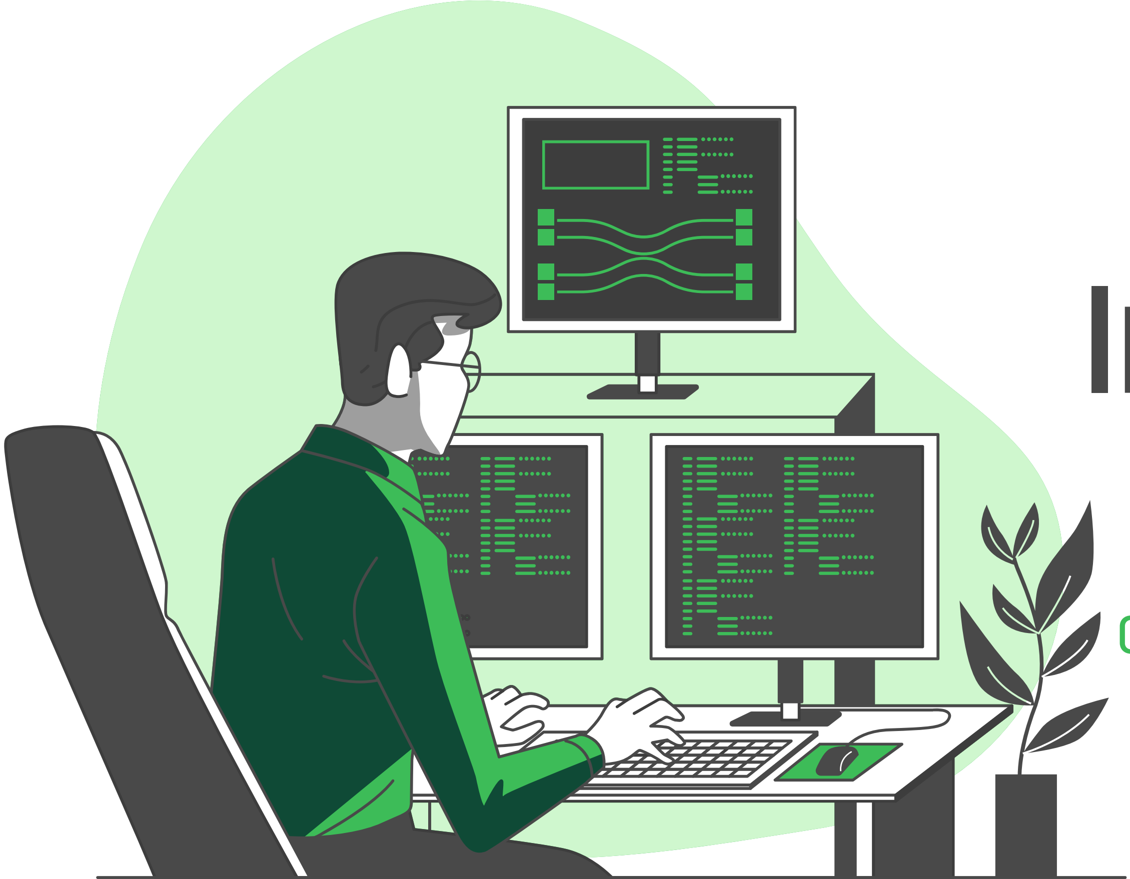
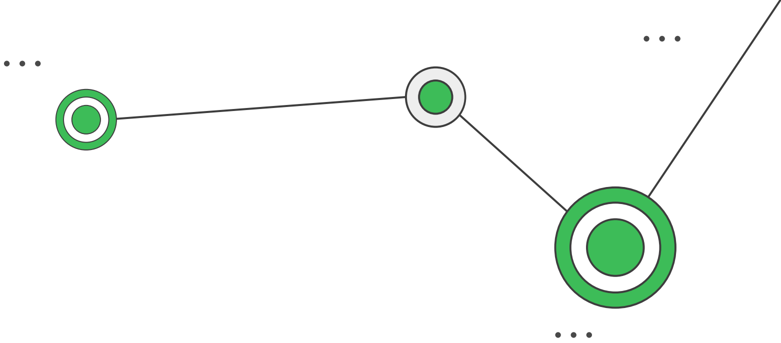


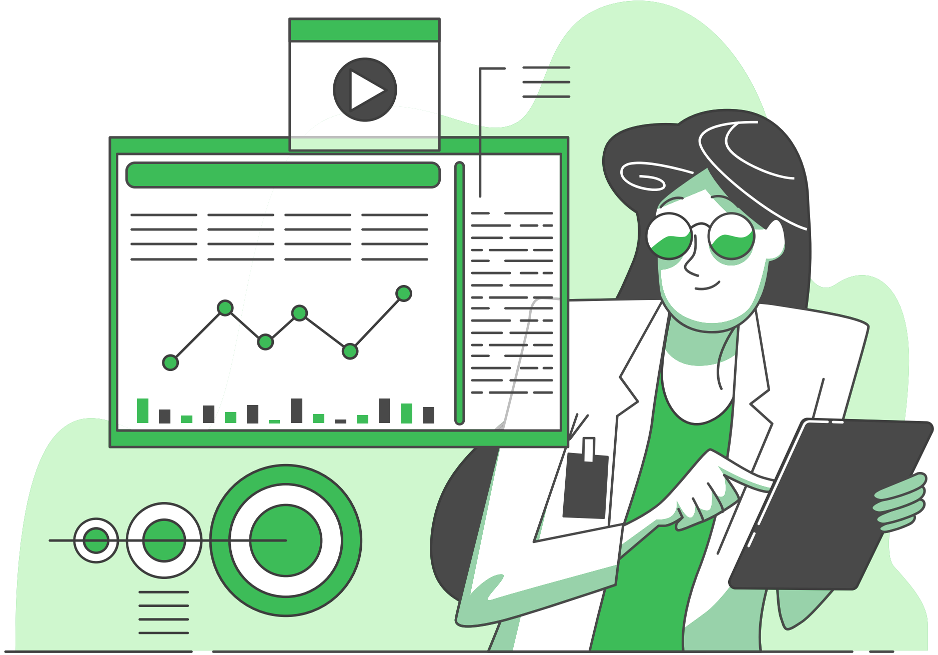


Introduction to Graphs

CS 251 - Data Structures and Algorithms

Note:
Slides complement the discussion in class

Table of Contents

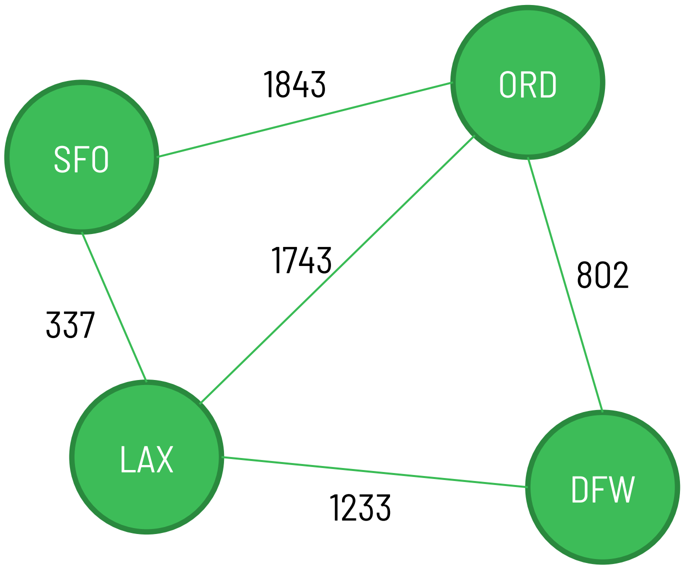


01

What's a Graph?

Practical examples and basic definitions

Graph



A graph G is a set of vertices V and a collection of edges E that connect a pair of vertices.

Notation: $G = \{V, E\}$ or $G(V, E)$

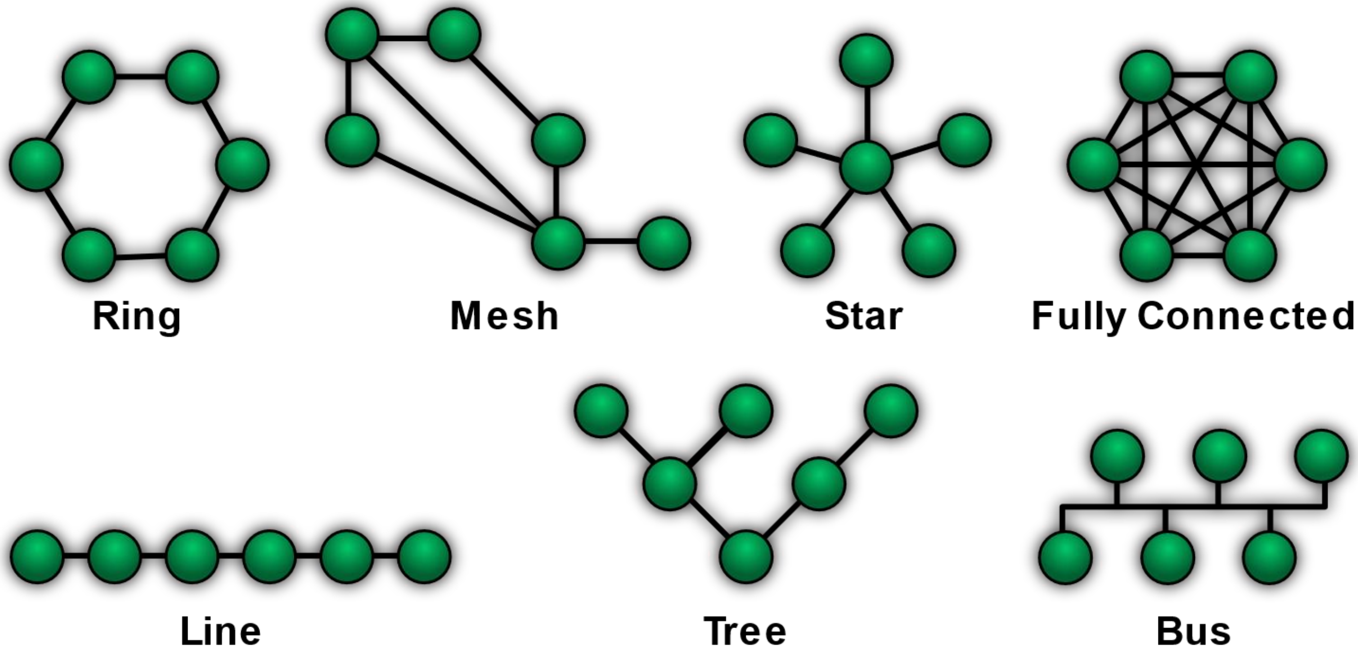
The vertices and edges store information.

https://commons.wikimedia.org/wiki/File:Konigsberg_bridges.png

Social Networks

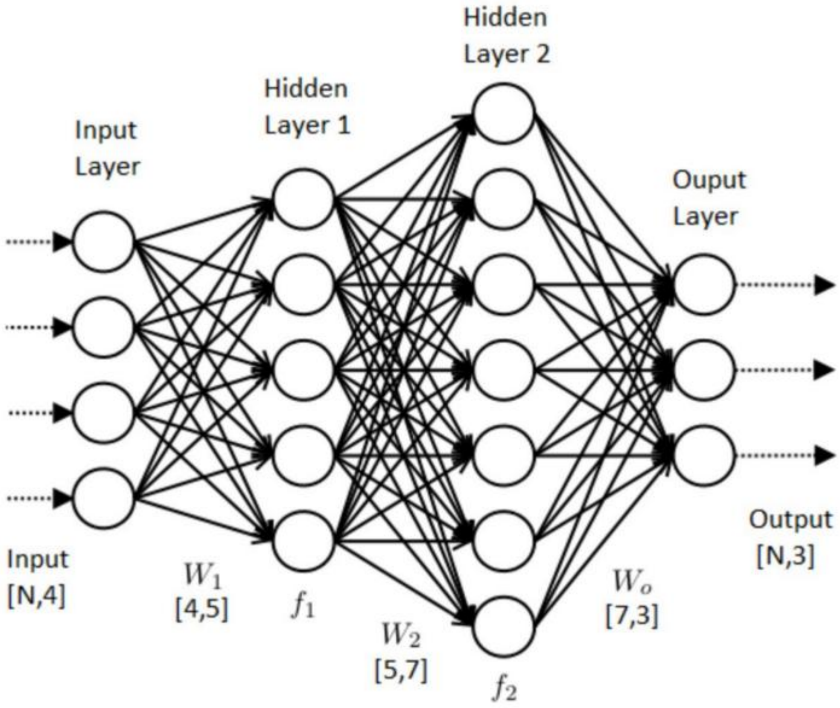
https://medium.com/analytics-vidhya/social-network-analytics-f082f4e21b16

Computer Network Topologies



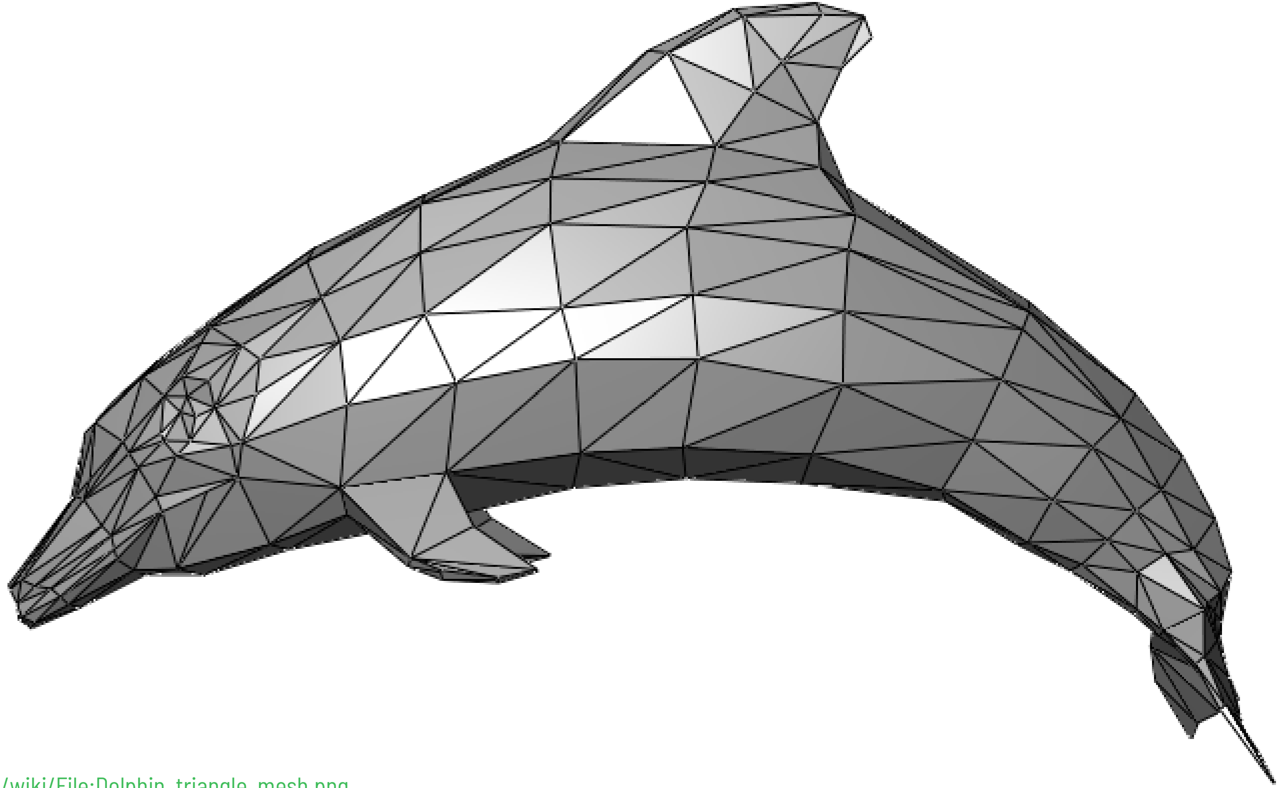
https://commons.wikimedia.org/wiki/File:NetworkTopologies.svg

Neural Networks



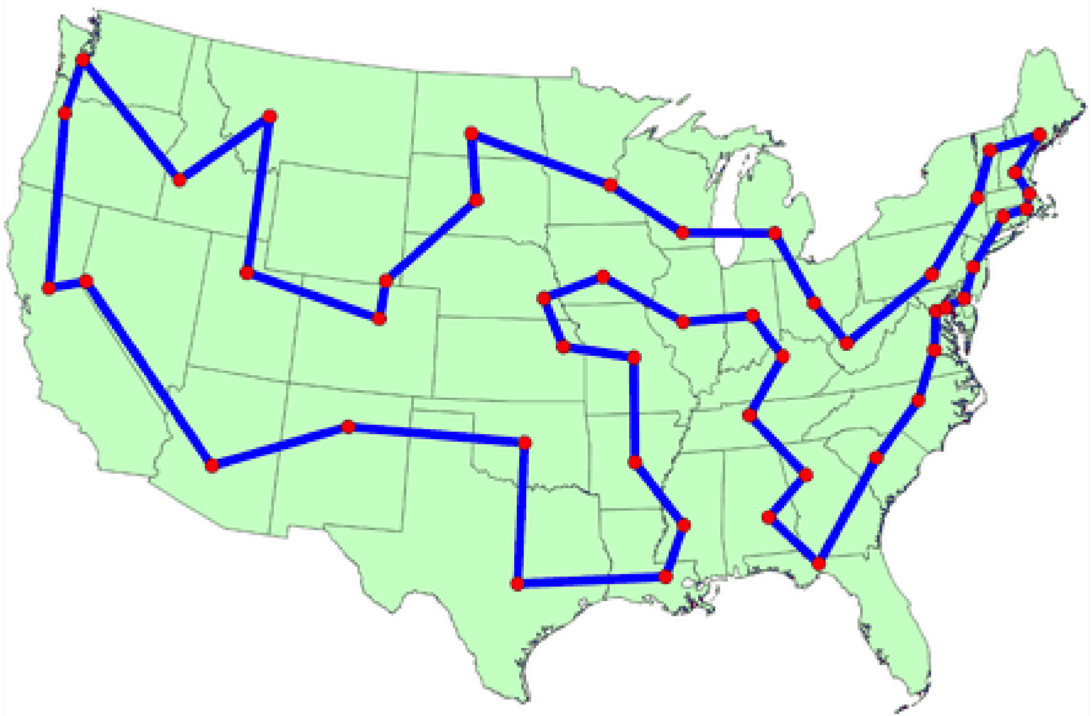
https://medium.com/coinmonks/the-artificial-neural-networks-handbook-part-1-f9ceb0e376b4

Polygon Mesh



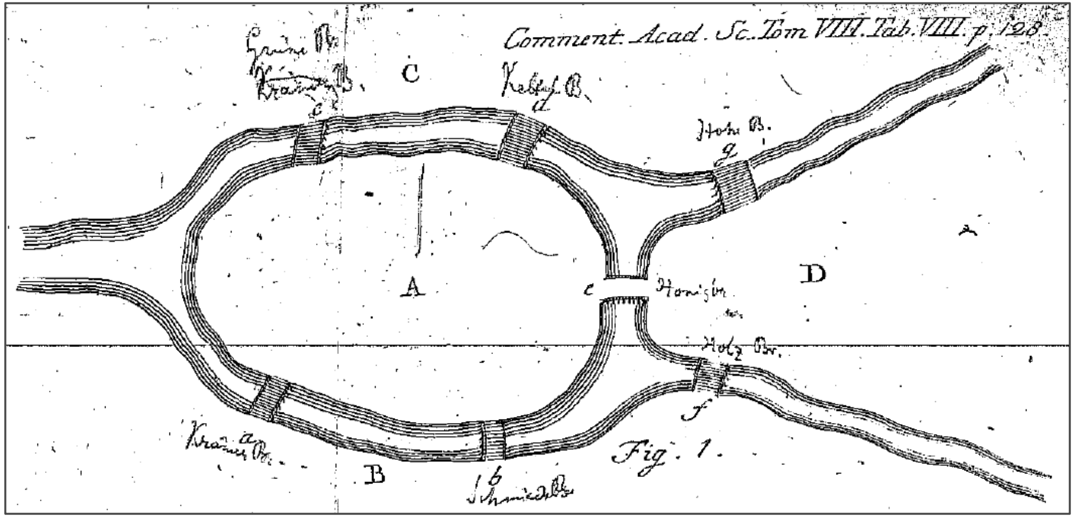
https://commons.wikimedia.org/wiki/File:Dolphin_triangle_mesh.png

Traveling Salesman Problem



https://optimization.mccormick.northwestern.edu/index.php/File:48StatesTSP.png

Seven Bridges of Königsberg

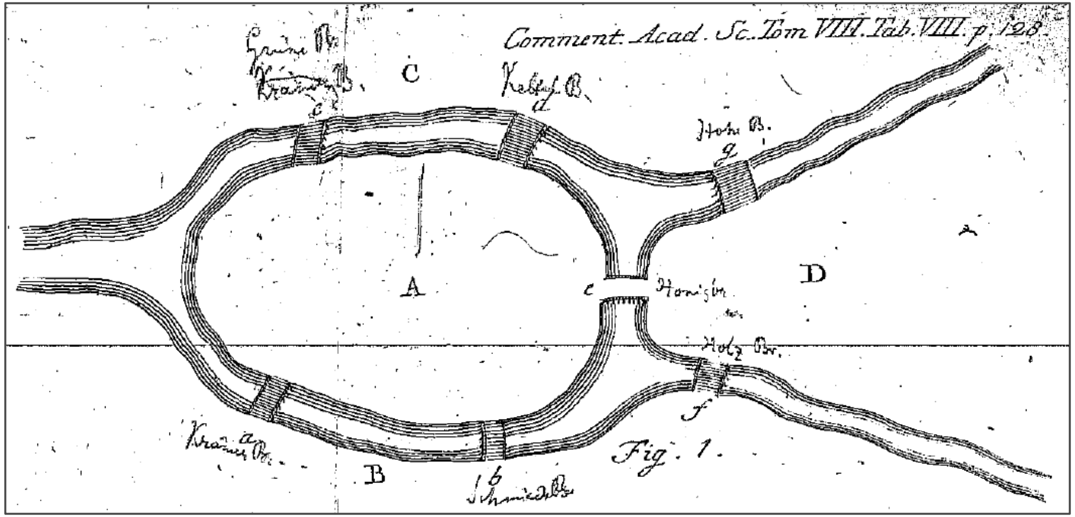


Problem: walk through the city that would cross each of those bridges once and only once.

Euler proved that for Königsberg (at that time) the problem had **no solution**.

Figure 1 from *Solutio problematis ad geometriam situs pertinentis* by Leonhard Euler, better known as The Seven Bridges of Königsberg.

Problem Modeling with Graphs



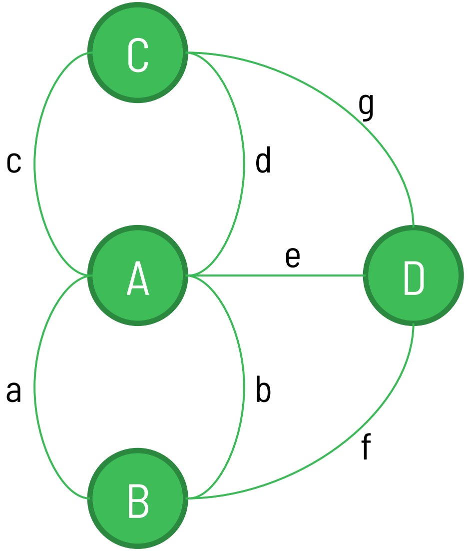


Figure 1 from *Solutio problematis ad geometriam situs pertinentis* by Leonhard Euler, better known as The Seven Bridges of Königsberg.

Edge Types

Directed

Ordered pair of vertices (u, v)

u: origin
v: destination
(Use arrows)

E.g., a flight

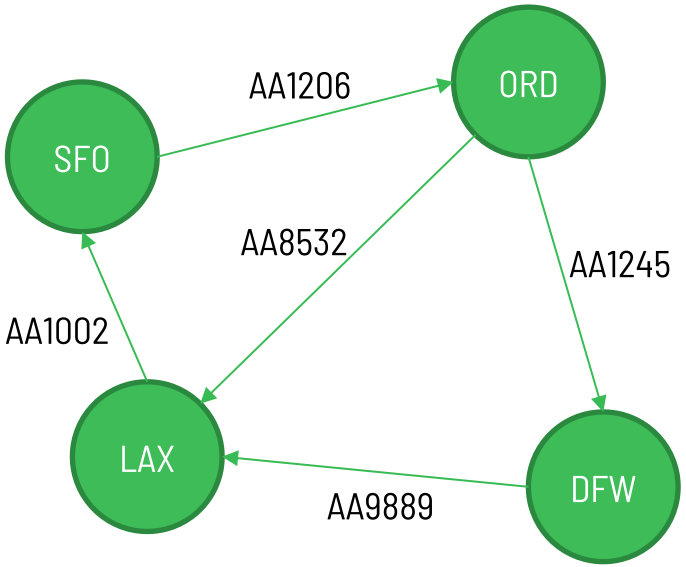
Undirected

Unordered pair of vertices (u, v)

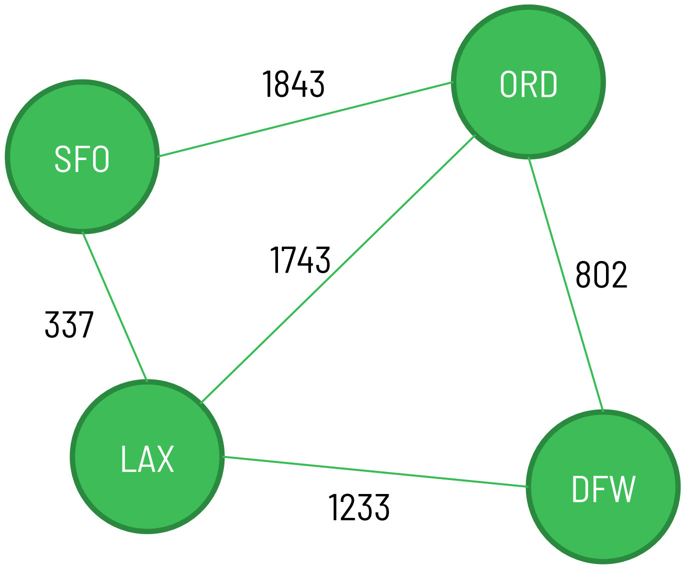
We move in both directions
(Use line segments)

E.g., a flight route

Edge Types



Directed Graph



Undirected Graph

Some Terminology

Endpoints (of an edge): the vertices connected by that edge (e.g., V and X are the endpoints of b).

Incident (on): when an edge touches a vertex (e.g., a is incident on V).

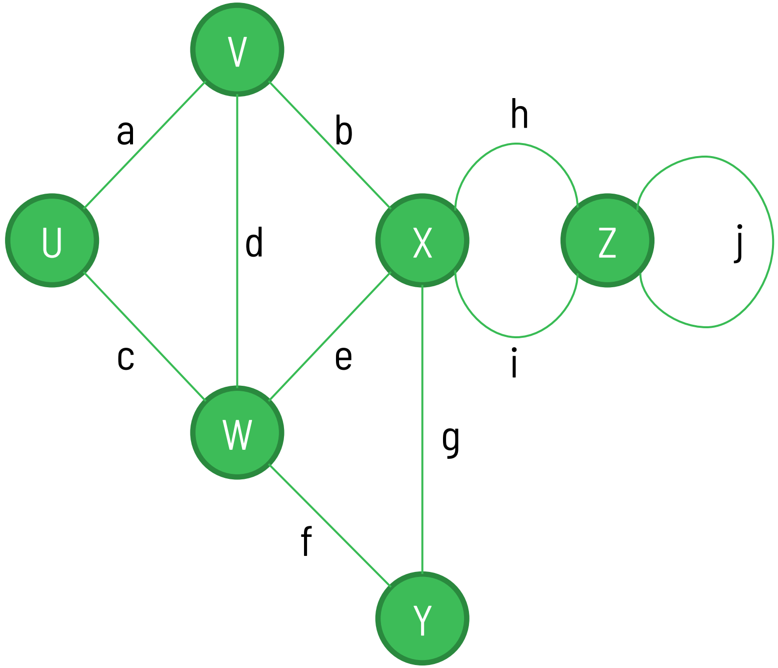
Adjacent (to): when two vertices are connected by a single edge (e.g., V and X are adjacent).

Parallel edges: edges with the same endpoints (e.g., h and i).

Self-loop: an edge that starts and ends at the same vertex (e.g., j).

Degree (of a vertex): the number of edges incident on that vertex (e.g., the degree of V is 3). **NOTE:** self-loops count twice towards degree (e.g., degree of Z is 4).

Simple graph: a graph with no parallel edges or self loops.



Handshaking Theorem

Let $G = (V, E)$ be an undirected graph. What is the relationship between the sum of the degrees of all the vertices and the number of edges?

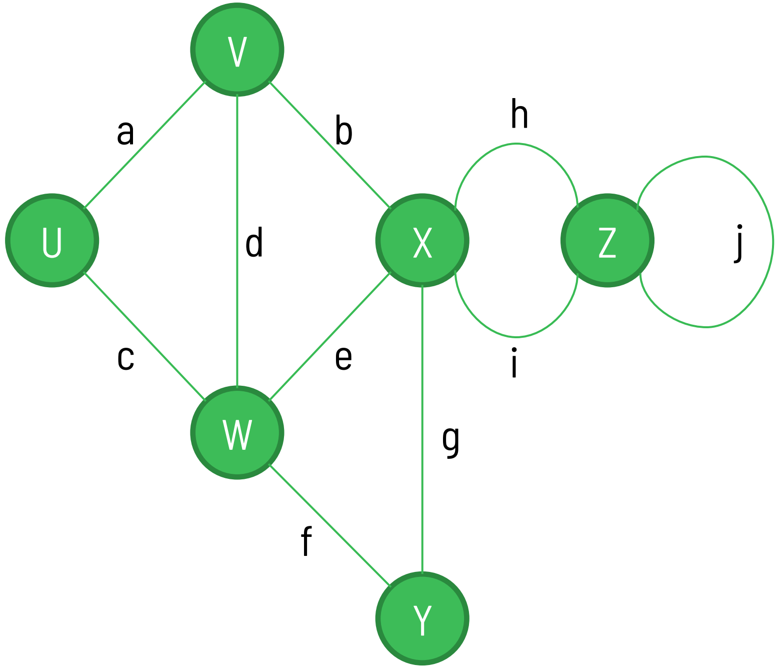
$$\sum_{v \in V} \deg(v) = 2 \cdot |E|$$

Proof rationale: each edge will be counted twice, once per endpoint.

$$\sum_{v \in V} \deg(v) = \deg(u) + \deg(v) + \deg(w) + \deg(x) + \deg(y) + \deg(z)$$

$$\sum_{v \in V} \deg(v) = 2 + 3 + 4 + 5 + 2 + 4 = 20$$

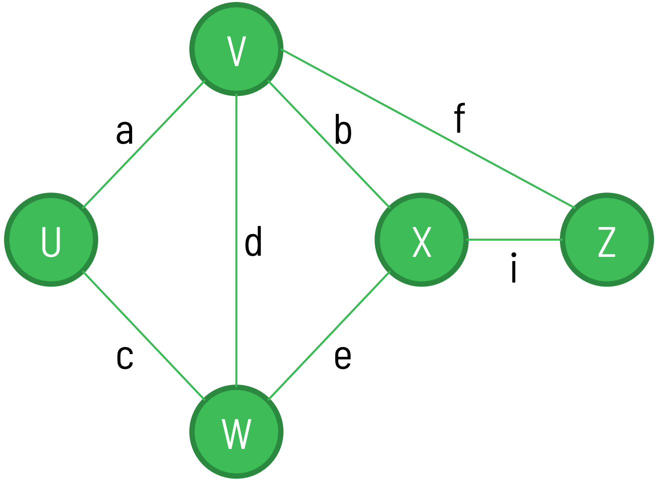
$$|E| = 10 \Rightarrow 2 \cdot |E| = 2 \cdot 10 = 20$$



Maximum Degree

Let $G = (V, E)$ be a simple, undirected graph. What is the maximum possible degree of any vertex?

The maximum possible degree of any vertex is $|V| - 1$ because at most a vertex can be connected to all the other vertices.

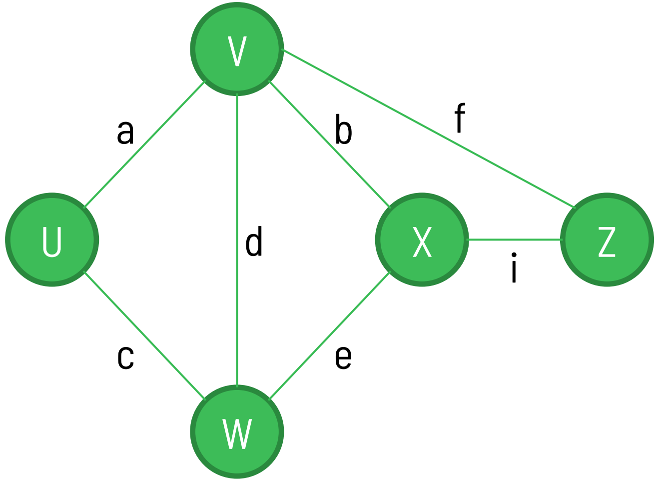


Edge Count

Let $G = (V, E)$ be a simple, undirected graph. What is the maximum possible number of edges in terms of V?

$$|E| \leq \frac{|V|(|V| - 1)}{2}$$

It follows from the previous property. There can be $|V|(|V| - 1)$ edges incident on each of the V edges, but we must divide by 2 since we count each edge twice.

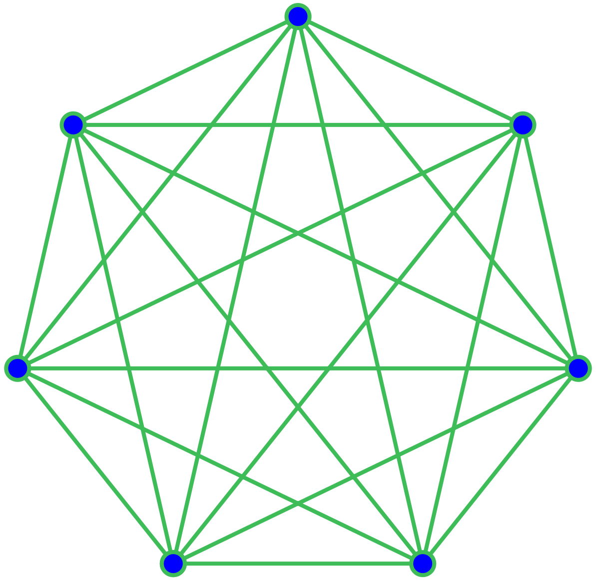


Complete Graph

A Complete Graph is a simple undirected graph in which each pair of distinct vertices is connected by a unique edge.

Properties:

- Each vertex is of degree $|V| - 1$
- $|E| = \frac{|V|(|V|-1)}{2}$



By David Benbennick - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=511711

More Terminology

Path: sequence of vertices connected by edges (e.g., {V, X, Z} and {U, W, X, Y, W, V}).

Simple path: a path with no repeated edges or vertices.

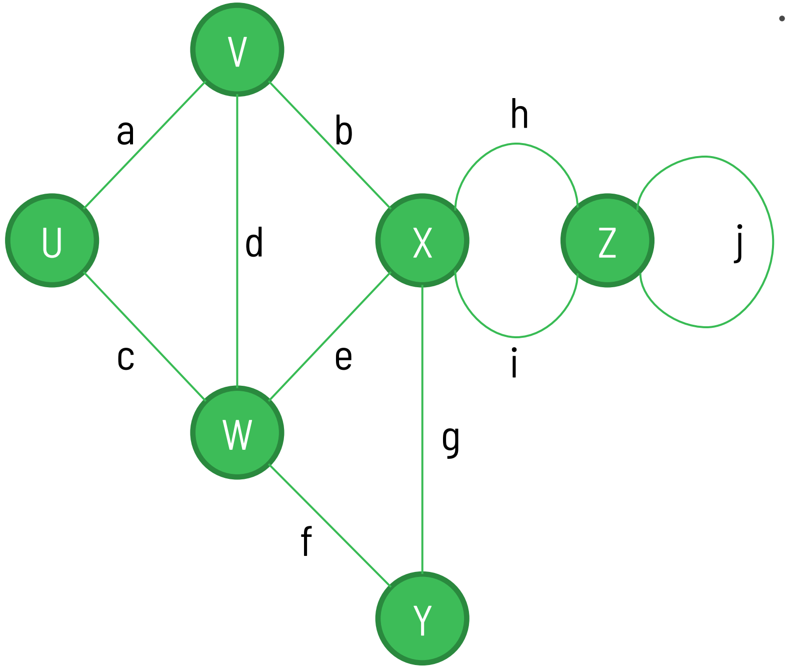
Cycle: a circular sequence of vertices connected by edges (e.g., {V, X, Y, W, U, V} and {U, W, X, Y, W, V, U})

Simple cycle: a cycle that has no repeated edges and no vertices (except the first and last, making it a cycle).

Length (of a path or cycle): the number of edges included in the path or cycle.

Subgraph: a subset of a graph's edges (and associated vertices)

Connected graph: a graph where there exists a path connecting any two pair of vertices.



Speaking of Paths

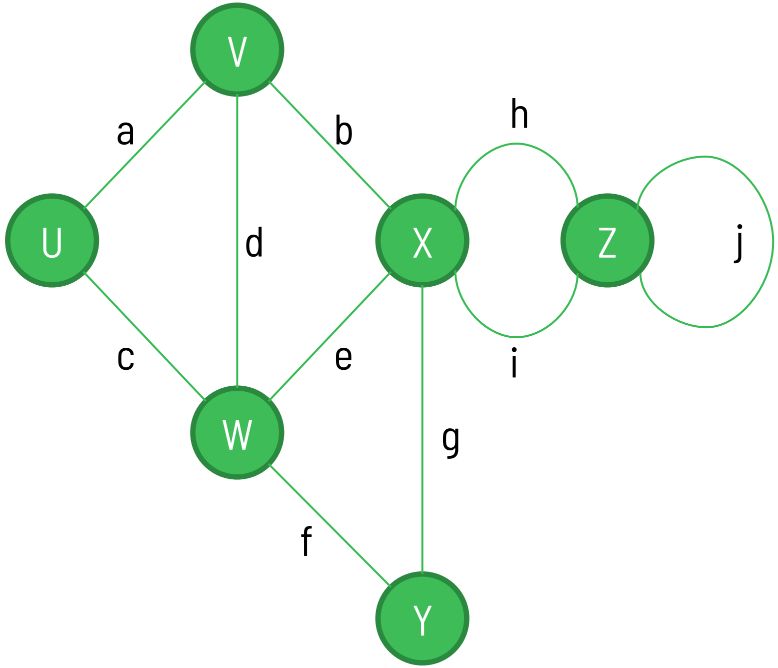
Euler Path: (aka. Eulerian Path or Eulerian Trail) A path that visits every edge of the graph exactly once.
E.g., {(V, U), (U, W), (W, V), (V, X), (X, W), (W, Y), (Y, X), (X, Z), (Z, Z), (Z, X)}

Euler Cycle: A Euler Path that starts and ends on the same vertex.

Hamiltonian Path: (aka. Traceable Path) A path that visits every vertex of the graph exactly once.
E.g., {U, V, W, Y, X, Z}

Hamiltonian Cycle: A Hamiltonian Path that starts and end on the same vertex.

Note: Euler/Hamiltonian Paths/Cycles are crucial for the definition and understanding of some problems in Math and Computer Science.



02

Representation

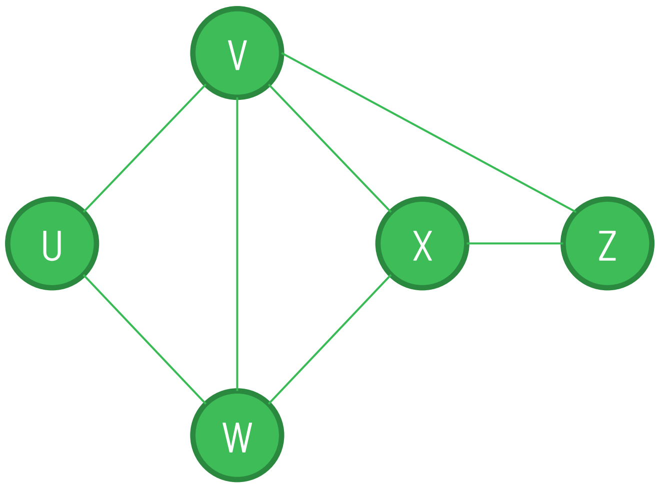
How computers work with graphs

Edge List

Represent the graph as a list of edges denoted by their endpoints.

E.g.,
{(U, V), (U, W), (V, X), (W, X), (V, W), (V, Z), (X, Z)}

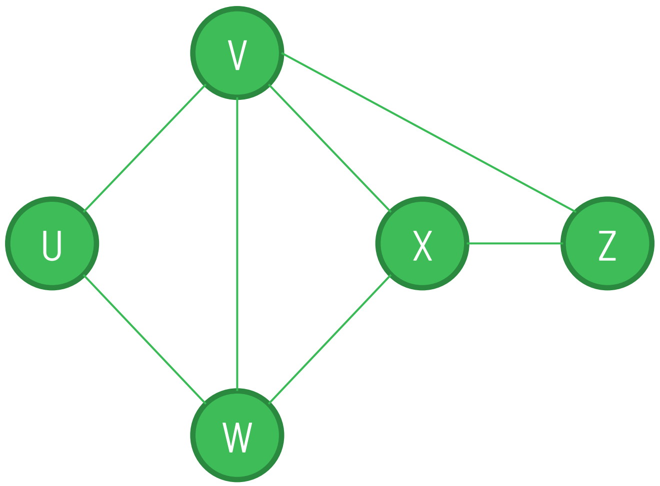
Space: $O(|E|)$
Add edge: $O(1)$
Check if two vertices are adjacent: $O(|E|)$
Iterate through the vertices adjacent to a vertex: $O(|E|)$



Adjacency Matrix

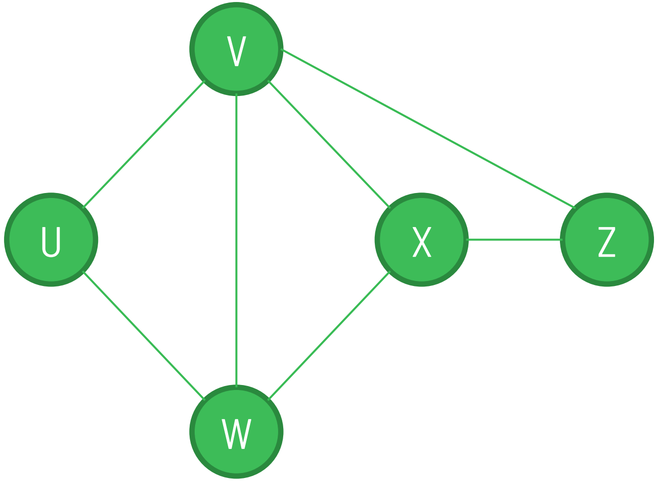
Represent the graph as a matrix where the columns are vertices, and the rows are vertices.

Each cell is either 0 or 1, depending on whether there is an edge between the respective vertices



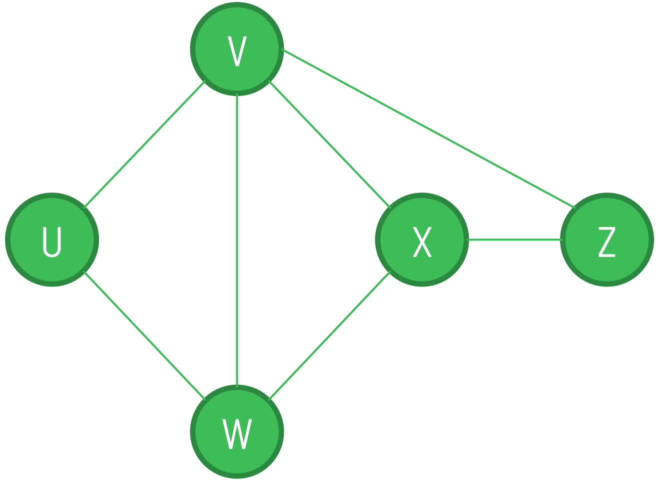
Adjacency Matrix

	U	V	W	X	Z
U	0	1	1	0	0
V	1	0	1	1	1
W	1	1	0	1	0
X	0	1	1	0	1
Z	0	1	0	1	0



Adjacency Matrix

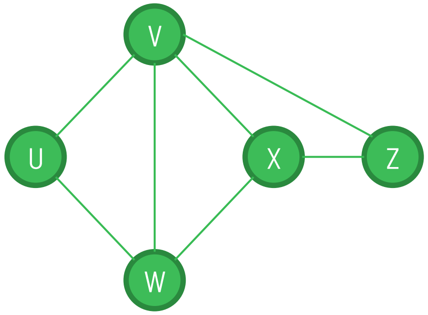
	U	V	W	X	Z
U	0	1	1	0	0
V	1	0	1	1	1
W	1	1	0	1	0
X	0	1	1	0	1
Z	0	1	0	1	0



Space: $O(|V|^2)$ (bad for Sparse Graphs $|E| \ll |V|^2$)
Add edge: $O(1)$
Check if two vertices are adjacent: $O(1)$
Iterate through the vertices adjacent to a vertex: $O(|V|)$ (slow if $\deg(v) \ll |V|$)

Number of Paths of Length k

Let G be a graph with adjacency matrix A and k be a positive integer. Then the matrix power A^k gives the matrix where $A_{i,j}$ counts the number of paths of length k between vertices v_i and v_j.



A	**U**	**V**	**W**	**X**	**Z**
U	0	1	1	0	0
V	1	0	1	1	1
W	1	1	0	1	0
X	0	1	1	0	1
Z	0	1	0	1	0

A^3	**U**	**V**	**W**	**X**	**Z**
U	2	6	5	3	3
V	6	6	7	7	6
W	5	7	4	7	3
X	3	7	7	4	5
Z	3	6	3	5	2

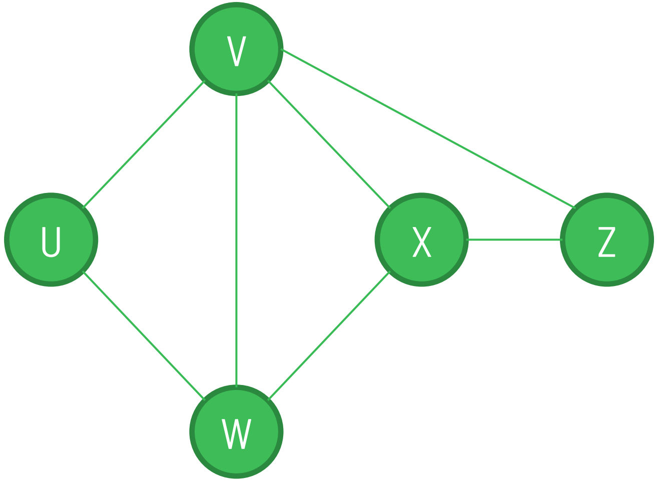
Wolfram Alpha: {{0,1,1,0,0},{1,0,1,1,1},{1,1,0,1,0},{0,1,1,0,1},{0,1,0,1,0}}^3

https://www.math.utah.edu/~gustafso/s2017/2270/projects-2017/dylanJohnson/Dylan%20Johnson%20Graph%20Theory%20and%20Linear%20Algebra.pdf

Adjacency List

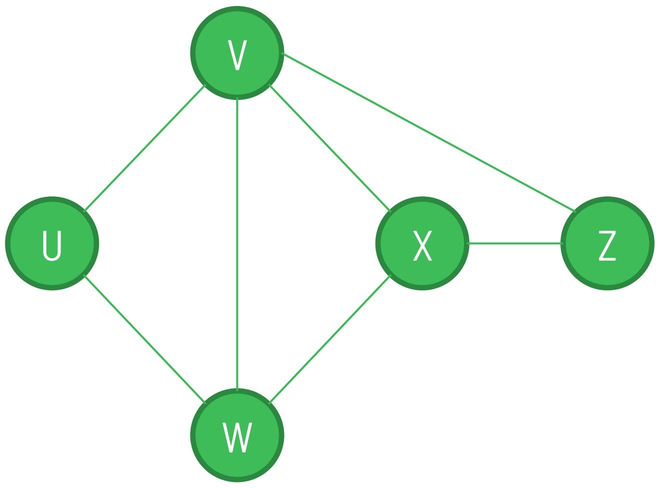
Represent the graph as an array of adjacency lists, where each list indicates the vertices adjacent to one of the vertices.

U	V, W
V	U, W, X, Z
W	U, V, X
X	V, W, Z
Z	V, X



Adjacency List

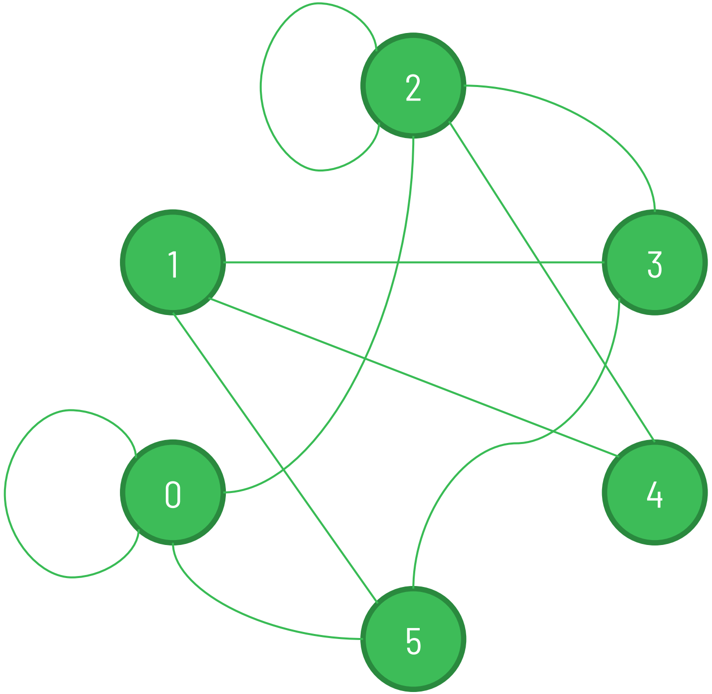
U	V, W
V	U, W, X, Z
W	U, V, X
X	V, W, Z
Z	V, X



Space: $O(|V| + |E|)$ (good for Sparse Graphs)
Add edge: $O(1)$ (NOTE: $O(\deg(v))$ if parallel edges not allowed)
Check if two vertices are adjacent: $O(\deg(v))$
Iterate through the vertices adjacent to a vertex: $O(\deg(v))$

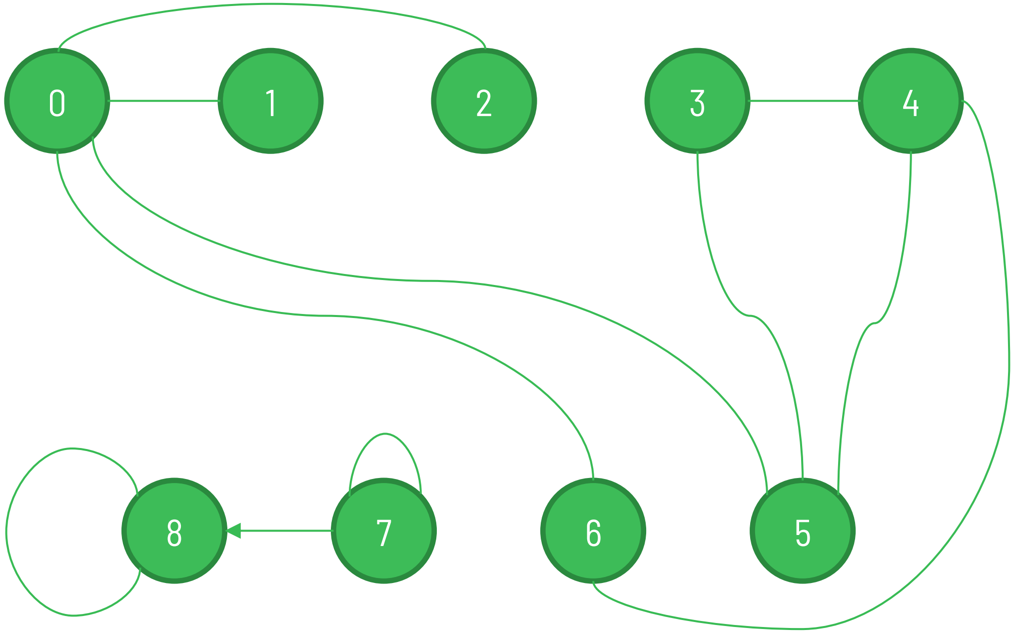
What does this graph look like?

	0	1	2	3	4	5
0	1	0	1	0	0	1
1	0	0	0	1	1	1
2	1	0	1	1	1	0
3	0	1	1	0	0	1
4	0	1	1	0	0	0
5	1	1	0	1	0	0



What does this graph look like?

0	6, 2, 1, 5
1	0
2	0
3	5, 4
4	5, 6, 3
5	3, 4, 0
6	0, 4
7	8, 7
8	8



Finis Est

Do you have any questions?