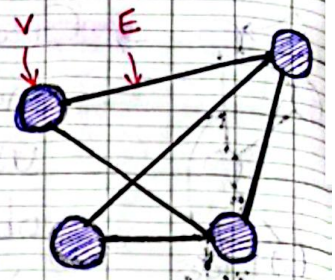
Graph Theory

Graphs in Graph Theory



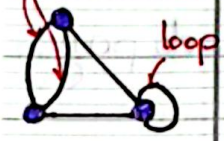
- Collection of nodes and edges
- Denoted by $G = (V, E)$ → edges (links), → vertices (points)
- Size: $n = |V|$, $m = |E|$

$|V| = 4$ $|E| = 5$

Vertex / Node
- basic element
- drawn as "node" or "dot".
- denoted by $V(G)$, V, or V_G
- represent object or entity

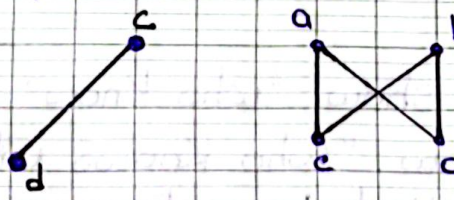
Edge / Arcs
- Set of 2 elements
- drawn as a line connecting 2 vertices called end vertices or endpoints
- denoted by $E(G)$, E or E_G
- represent a relationship

Neighborhood
- For any node v the set of nodes it's connected to via an edge is called neighborhood
- denoted by $N(v)$

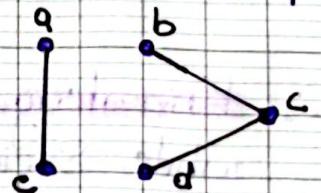


Simple Graph: no loops, no multiple edges

Connected Graphs
- There exists at least one path between two vertices

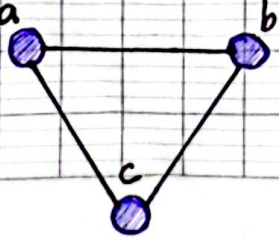


Disconnected
- otherwise (separate, unconnect

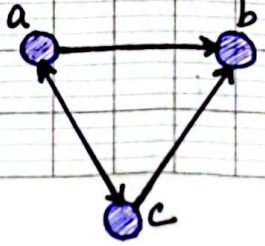


Edge Types

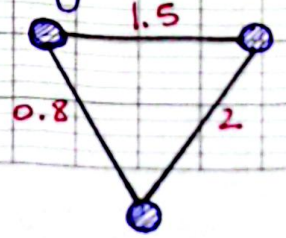
Undirected
- 2-way, symmetric link



Directed
- 1-way, assymetric link, source & target



Weighted
- with associated weight (numerical val

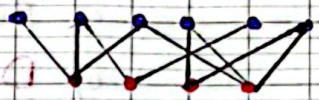


→ Graph Classes

1) Cycle Graph (C_n)
- Single cycle connecting all vertices
- $V = n$, $E = n$

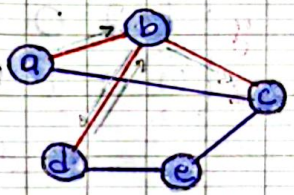
2) Complete Graphs
- Every pair of vertices is connected by one edge
- $E = \frac{n(n-1)}{2}$

3) Bipartite Graphs
- Vertices divided into two disjoint sets
- No edges within the same set



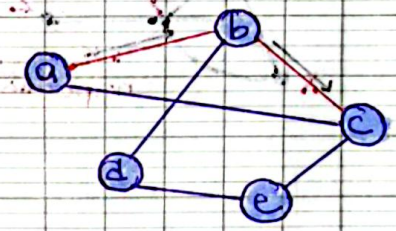
→ Graph Traversal Concepts

1) Walk
- From v_0 to v_i
- Sequence of vertices and edges
- Vertices and edges can be repeated



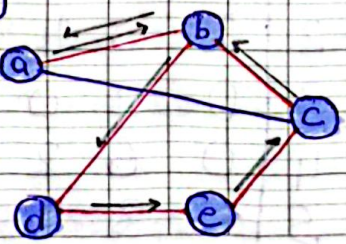
↳ ex: ⟨a, b, d, b, c⟩

2) Path
- From v_0 to v_i
- Sequence of vertices
- No vertex is repeated



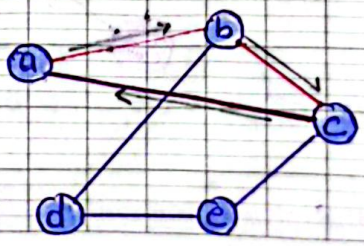
↳ ex: ⟨a, b, c⟩

3) Circuit
- From v_0 to v_0
- walk that starts and end at the same vertex
- can repeate vertices and edges



↳ ex: ⟨a, b, d, e, c, b, a⟩

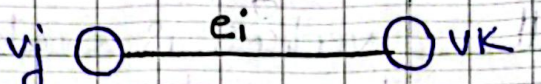
4) Cycle
- From v_0 to v_0
- Circuit with no repeated vertices (except start/end)
- Start and end at the same vertex



↳ ex: ⟨a, b, c, a⟩

→ Graph Representation

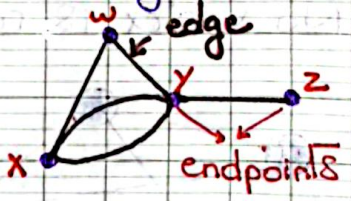
• Notations



↳ vertices (v_j and v_K) said to be **adjacent**
↳ edge (e_i) said to be **incident** upon v_j
↳ **Degree** of v_K is the nb. of edges incident upon v_K (denoted $d(v_K)$)

• 3 ways to represent a graph

1) Adjacency matrix A(G)

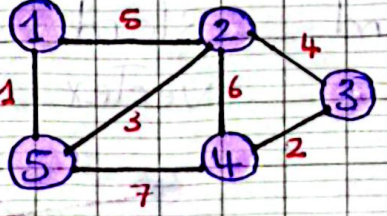


$$\begin{array}{c} \\ w \\ x \\ y \\ z \end{array} \begin{array}{c} \begin{array}{cccc} w & x & y & z \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

|V| x |V| matrix

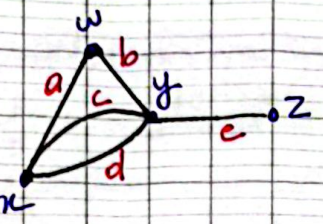
number of edges in G with endpoints $\{v_i, v_j\}$

↳ Adjency matrix for a weighted graph
• instead of adding nb. of edges, we add the weight of the edge, if no weight we add 0



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 5 & 0 & 0 & 1 \\ 5 & 0 & 4 & 6 & 3 \\ 0 & 4 & 0 & 2 & 0 \\ 0 & 6 & 2 & 0 & 7 \\ 1 & 3 & 0 & 7 & 0 \end{bmatrix} \end{array}$$

2) Incidence Matrix



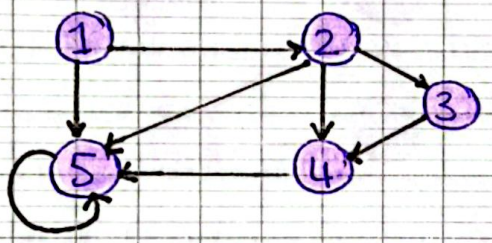
endpoints ↓ | a | b | c | d | e ← edges

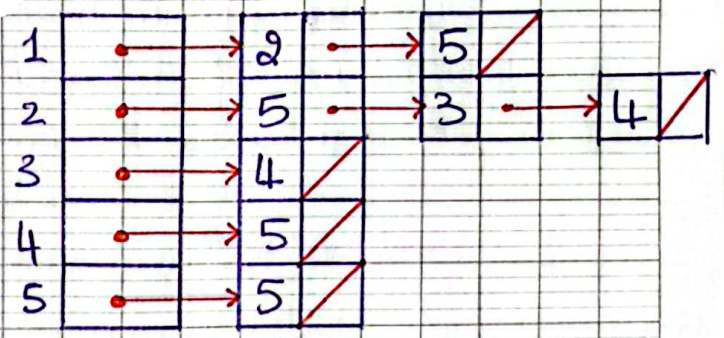
$$\begin{array}{c} \\ w \\ x \\ y \\ z \end{array} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- $|V| \times |E|$ matrix
- 1 if v_i is an endpoint of e_j , 0 otherwise

3) Adjency List

- Array of $|V|$ elements, one for each vertex in V



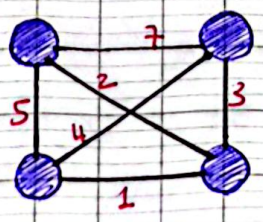


→ Spanning Tree

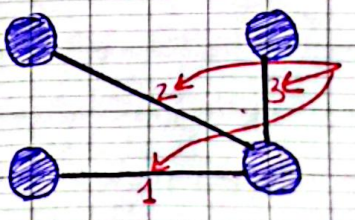
- Subgraph that contains all the vertices and is a Tree
- connected, acyclic, has exactly n-1 edges
- Graph can have manny spanning Trees

↳ Minimum Spanning Tree

- Spanning tree with minimum total edge weight



Complete graph



Minimum Spanning Tree