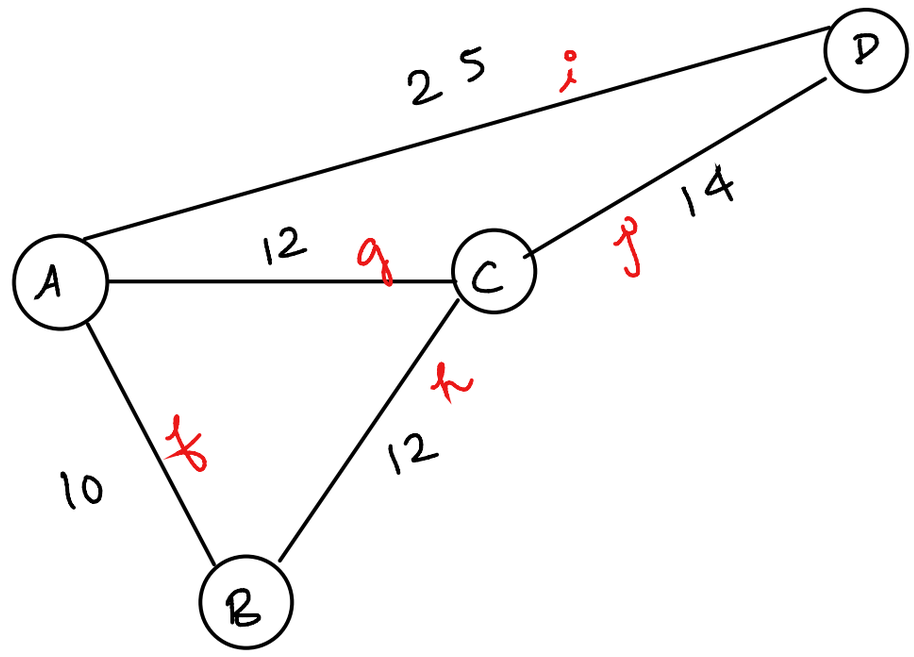
# Graphs

$G = (V, E)$

$V$ → Set of Vertices

$E$ → Set of edges

Examples




$V = \{ A, B, C, D \}$

$E = \{ f, g, h, i, j \}$

Vertices: cities

Edges : Roads connecting the cities

Any network

road, railways,

computer,

Edges :

Directed

Undirected

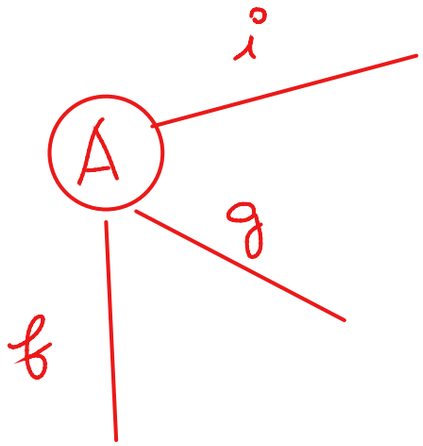
- End points of an edge : Two vertices between which the edge exists

end points of g : A , C

i : A, D

- Edges incident on a Vertex :

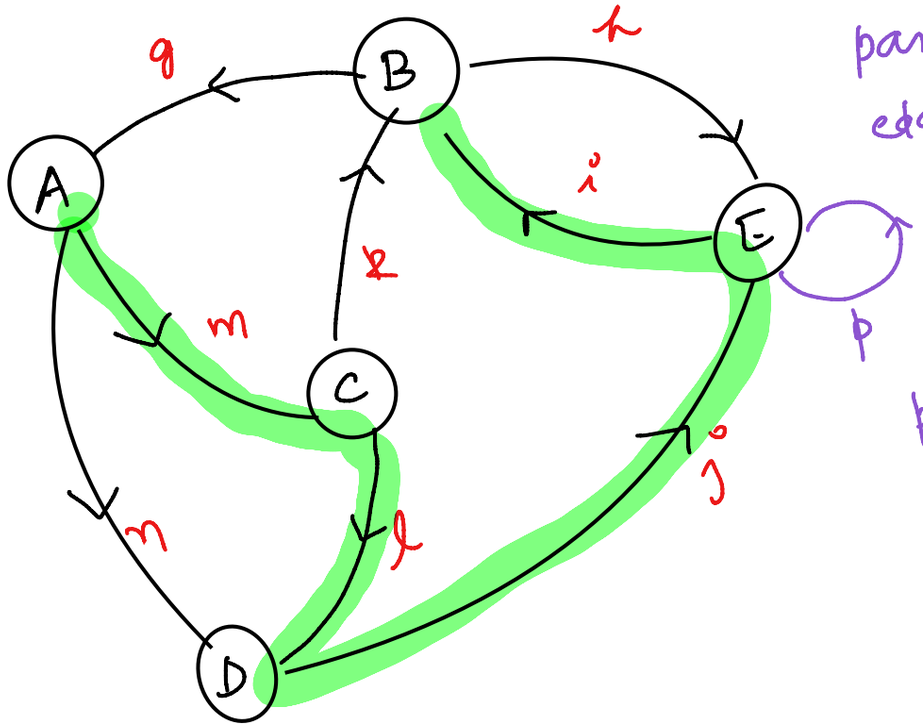
on A :




- Adjacent vertices : A, B and D are adjacent to C

A and C are adjacent to D

- Outgoing vs incoming edges

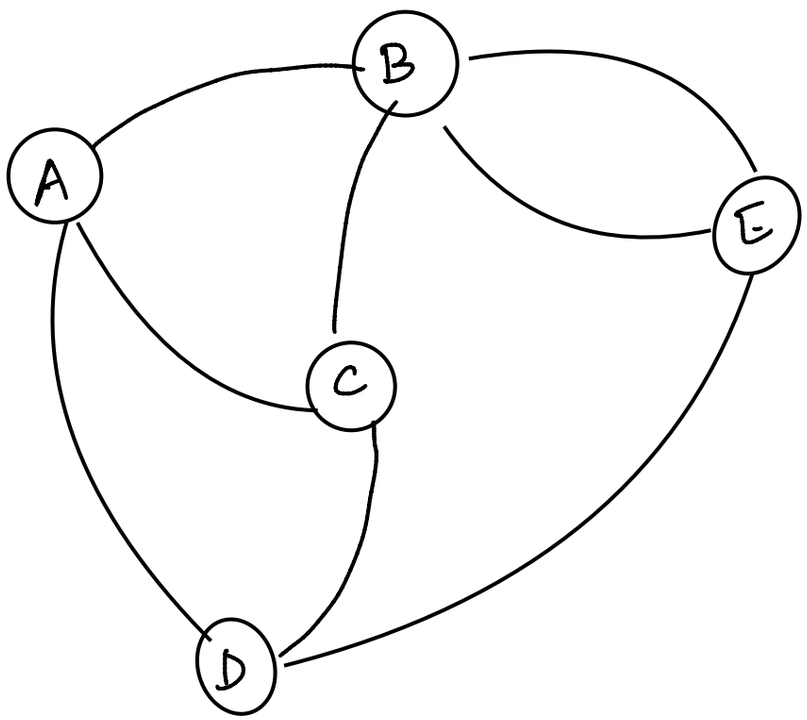



h, i are parallel edges

p: self loop

| | in degree | out degree |
|---|---|---|
| A | 1 | 2 |
| B | 2 | 2 |
| C | 1 | 2 |
| D | 2 | 1 |
| E | 2+1 | 1+1 |

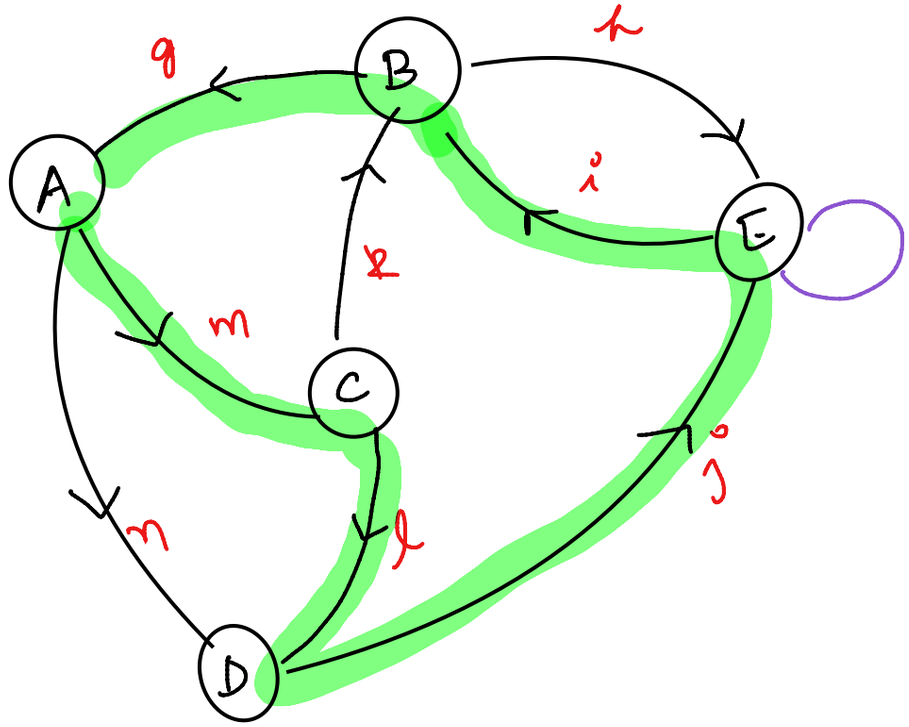
A m C l D j E i B




| | Degree |
|---|---|
| A | 3 |
| B | 4 |
| C | 3 |
| D | 3 |
| E | 3 |
| sum | 16 = 2 x No of Edges |

- Path: Sequence of vertex, edge, vertex, edge, . . . . , vertex

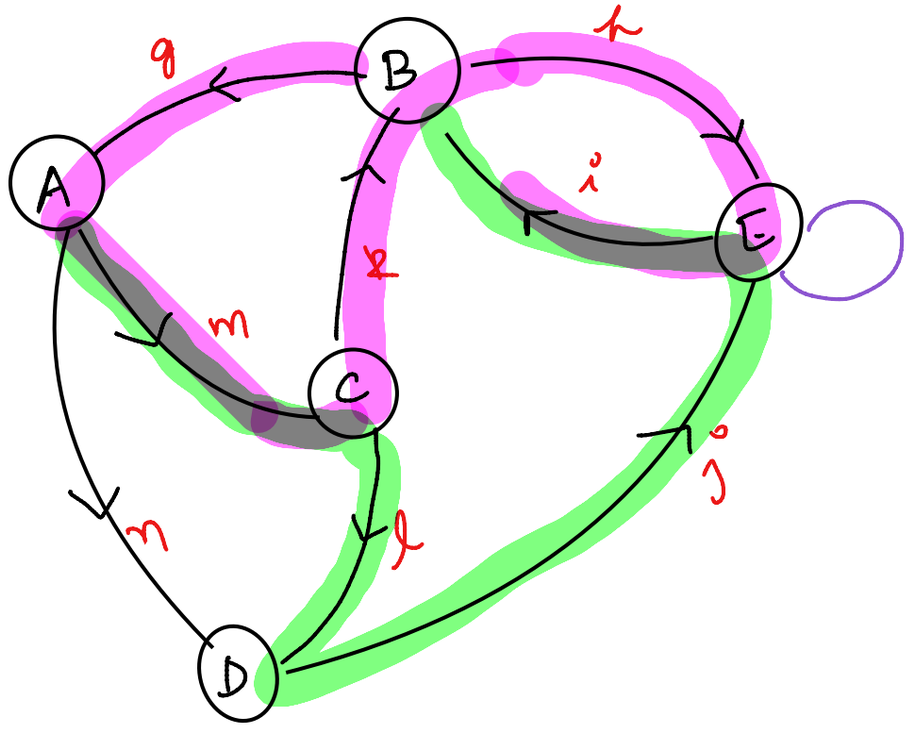
Simple Path: No element repeats




Simple Cycle: A m C l D j E i B g A

starts

ends with same vertex



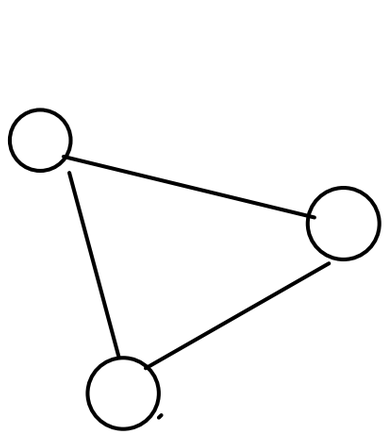

Not a simple cycle : A m C k B h E i B g A

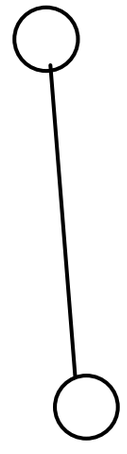
Reachability: A vertex V is reachable from another vertex U if there exists a path from U to V


In undirected graph V is reachable from U

$\Rightarrow$ U is reachable from V

Connected Graph: Pick any two vertices then there is a path between the two.





Subgraph: Formed by $G_S = (V_S, E_S)$

of

Graph $V_S \subseteq V$ , $E_S \subseteq E$

$G = (V, E)$

Tree: Connected Graph with no cycles

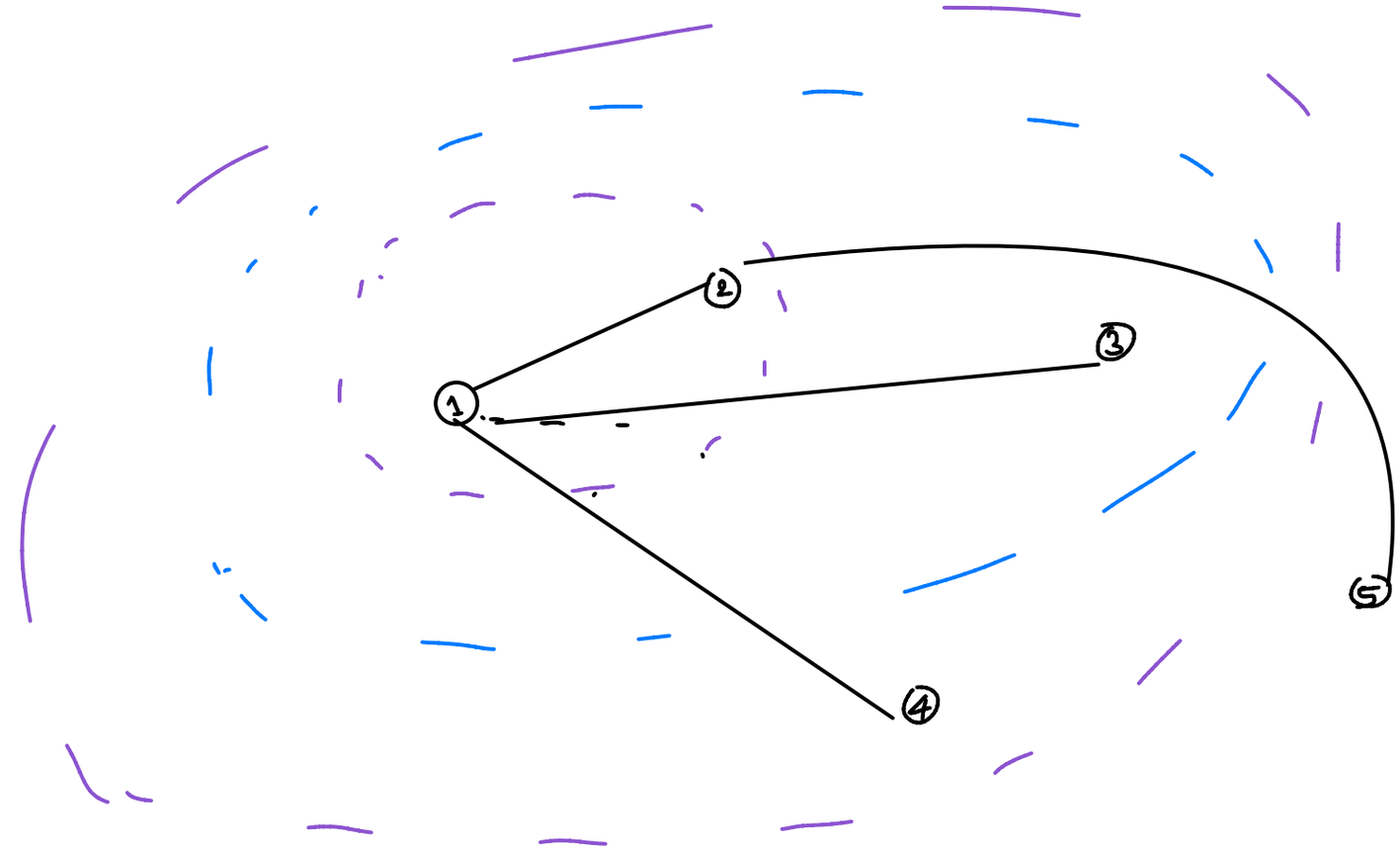
Forest: Collection of trees.

maximum number of edges in an undirected graph

$$n C_2 = \frac{n(n-1)}{2}$$

Let $G$ be an undirected graph with $n$ vertices and $m$ edges.

- $G$ is connected $m \geq n-1$




- $G$ is a tree $m = n-1$
- $G$ is a forest $m \leq n-1$

Graph ADT

- Vertex Objects

  element() (name of vertex)
  or some satellite data

- Edge objects

  source()

  destination()

  cost()

- overall Graph Object

3 ways of storing / representing

Edge List, Adjacency list / map, Adjacency matrix