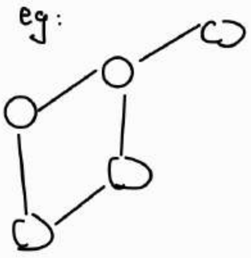
Graphs

(No specific structure)

- Nodes can be connected freely
- social media network (mutual connections etc.)

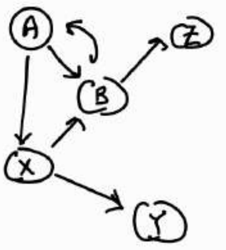
eg:



Types of Graphs

1. Directed

- Has direction specified

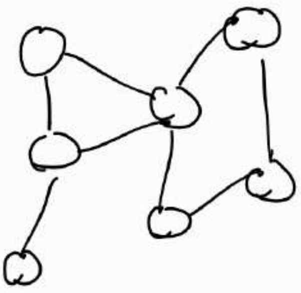


AXY

AXBZ

* cannot reverse dirn

2. Undirected

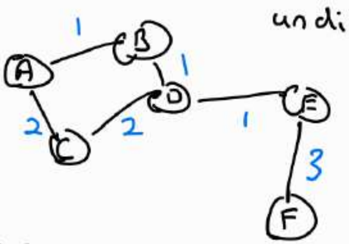


- No direction restriction
- can travel in any dirn

(May result in loop)

3. Weighted

- connections are weighted
- can be for directed or undirected



- quickest way to move between points

A → F

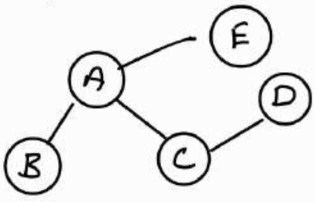
ACDEF : 8 pts

ABDEF : 6 pts (Better)

Graph Terminology

1. Vertex, plural = vertices : Any point on graph (i.e nodes)

vertex = nodes



2. Edge = connection between vertices

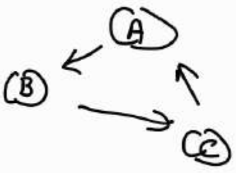
Edge set = {AB, AE, AC, CD}

Vertex set = {A, B, C, D, E}
└ order does not matter

3. Adjacency = 2 vertex connected together are adjacent

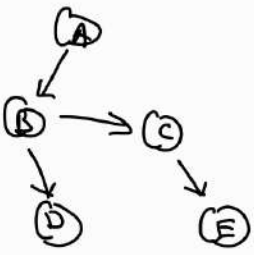
4. Cyclic graph : At least 1 cycle where start & end node same
(May enter infinite loop)

eg:



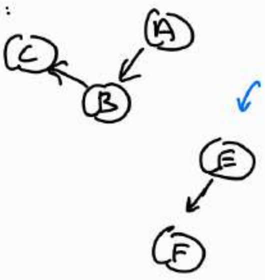
5. Acyclic graph : Non cyclic graph

eg:



6. Disconnected graph

eg:



eg: new user
(no connection)
cannot just erase

Depth - First search (utilize stack) (good for depth search)

- Avoid cyclic loop / repeated nodes exploration

1. Choose starting node & add to stack
2. Find all unexplored node & add to stack (order doesnt matter)
   - L if no more deeper nodes, pop node out of stack (move back)
     - repeat step 2.

* Due to using stack, no repeated search performed.

Breadth - First search (utilize queue) (Good for width search)

- process all adjacent nodes first instead of going into deeper nodes

1. Choose starting node & add to queue
2. Deque from queue & look at existing child, add childs to queue
   - L repeat step 2 until all explored

DFS & BFS complexity

$O(V+E)$, V = vertices, E = edges

cannot quantify with n since vertices & edges may vary.