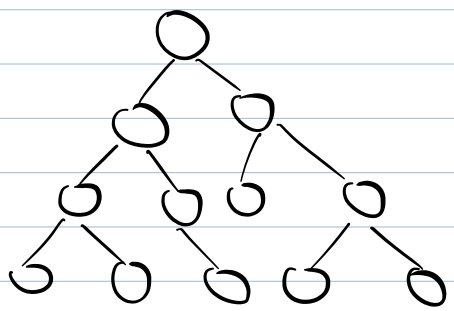
What is a graph ??

A network (any network)

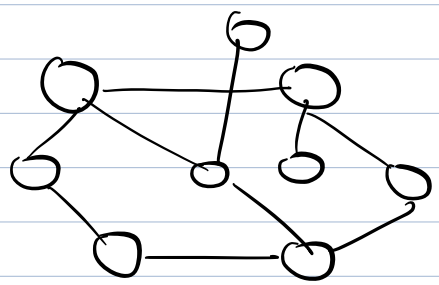
1) Nodes (vertices)
2) Edges

⇒ Graph is a collection of nodes & edges.

Tree



Graph



1) Tree is hierarchical data structure unlike graph.

2) No. of edges in a tree with N nodes = N-1
   ↳ no cycle.

Examples

1) Social Media
   - FB
   - Insta
   - Twitter
   - LinkedIn.

What is a node
You (User)

2) Website

3) GPS (Google Maps)

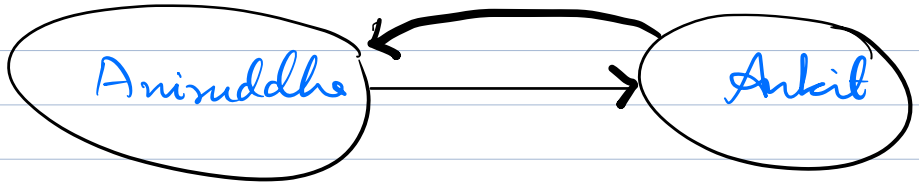
Node → Co-ordinate (Landmark)
Edges → Paths b/w landmarks.

4) Project Management Tool (Maven ↓ Java)

- Package 1
- Package 2
- Utility
- P3
- P4
- ⋮

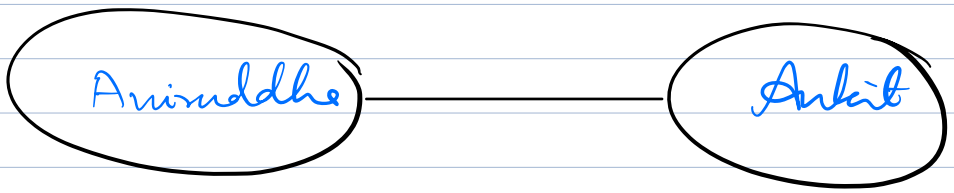
(Topological Sort)
DAG

Instagram.



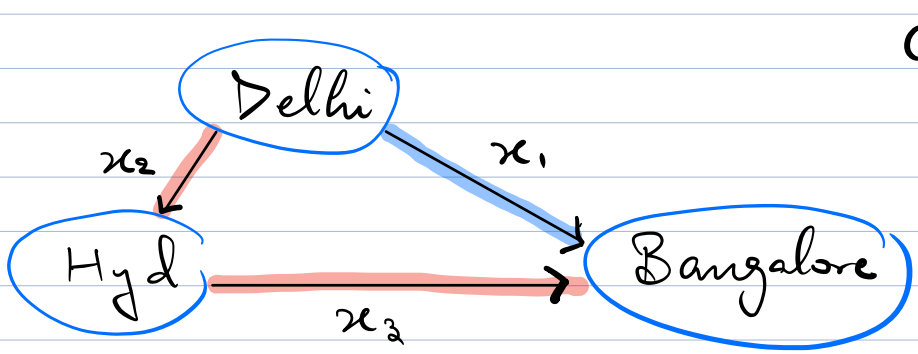
Directed Graph

Facebook



Undirected graph.

Flight Booking System



Weighted graphs.

Weight = 1 (Unweighted graph)

$\forall$ edges weight = $w > \frac{1}{}$ → Weighted.

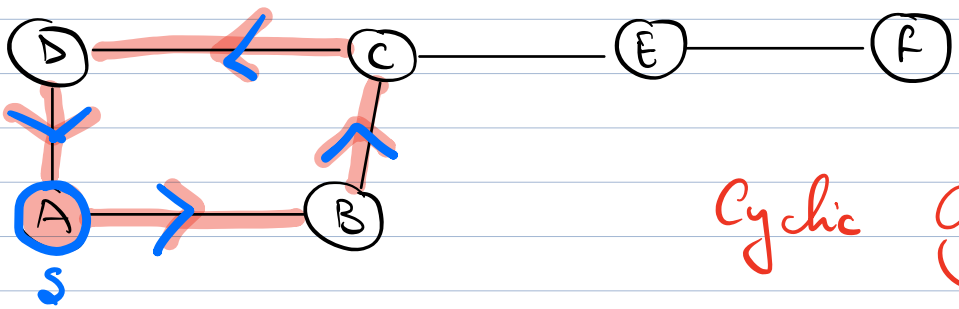
Classification

4

Directed / Undirected

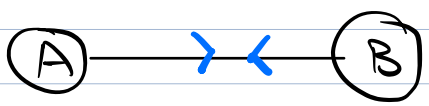
Weighted / Un-weighted.

1)



Cyclic Graph

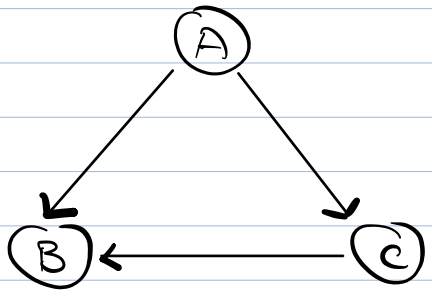
2)



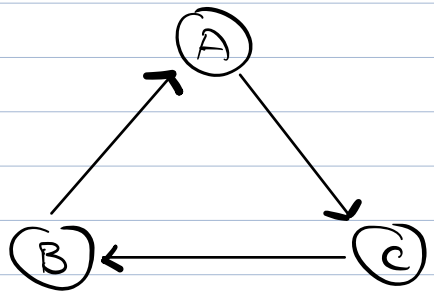
NO

AB
AB

3)

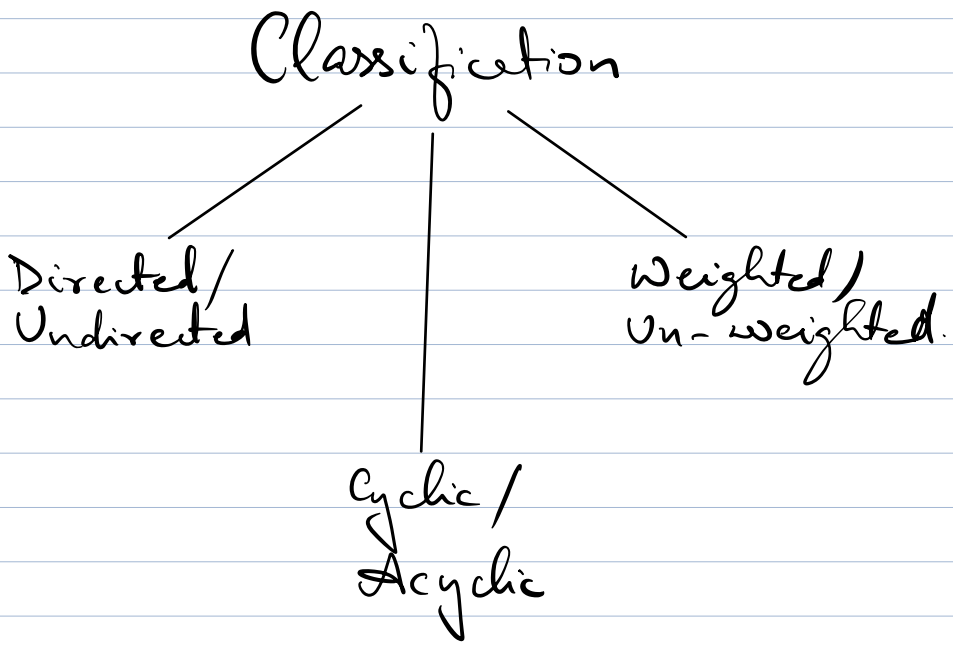


4)

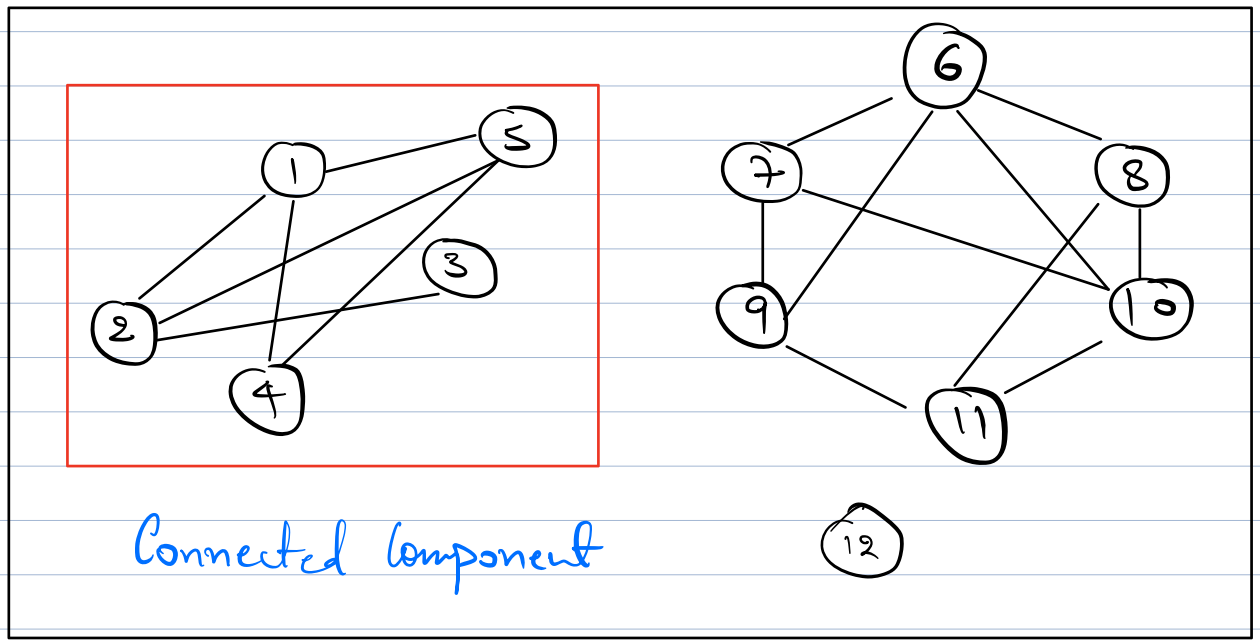


Yes

4)



Facebook.



1 Graph

Connected component (Undirected Graph) ⇒ A set of vertices in a graph where every vertex should be reachable from every other vertex.

Strongly Connected Component (Last session)

How Graph is given as an input??

(Collection of nodes & edges)

Undirected graph with N nodes & M edges.

Sol^{n}

→ 1st line : N M [1 to N]

#Nodes #Edges

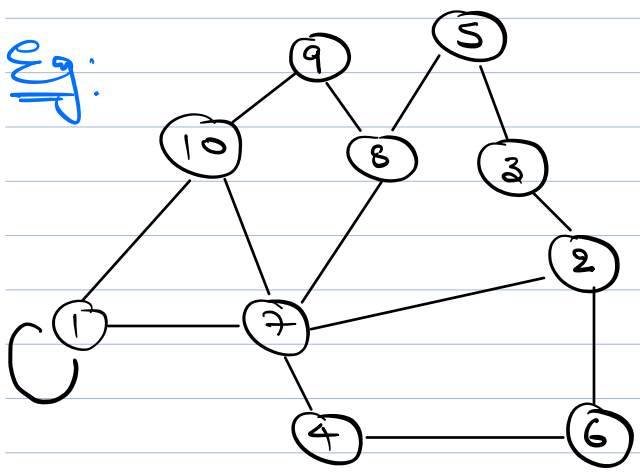
→ M lines

u v w

u & v : Edge b/w the nodes

w : Weight of this edge.

Eg:



N = 10
M = 14

1st → 10 _ 14

u		v		w
2	_	3	_	6
4	_	7	_	4
8	_	9		
2	_	7		
7	_	8		
10	_	1		
4	_	6		
5	_	8		
2	_	6		
10	_	9		
7	_	10		

Given :
1) Directed or undirected.
2) Weighted or un ..

Not Given
Cyclic or Acyclic.

3 — 5
7 — 1
1 — 1

★ Representation

1) Adjacency Matrix [V, E]

$$^{V}C_2 = \frac{(V)(V-1)}{2} = O(V^2)$$

V vertices → [0, N-1] = V×V

V vertices → [1 to N] = (V+1) × (V+1)

V	E
5	7
u	v
1	4
2	5
3	2
4	3
2	4
3	5
1	2

Mat[6][6]

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

Mat[i][j] ⇒ Information about the edge b/w the node i & node j

0 → No edge
1 → Edge

	Un Weighted	Weighted (u, v, w)
Undirected	If Mat[u][v] = 1 Then Mat[v][u] = 1	Mat[u][v] = w Mat[v][u] = w
Directed	Mat[u][v] = 1	Mat[u][v] = w

In general weights are non zero.

T.C. = O(E)
S.C. = O(V²)

Pros

1) Check if there is an edge b/w u & v
   ↳ O(1) (M[u][v] & M[v][u])

2) Add an edge b/w u & v.
   ↳ O(1)

(M[u][v] = 1 / w
M[v][u] = 1 / w (Undirected))

3) Remove an edge
↳ O(1)

Cons

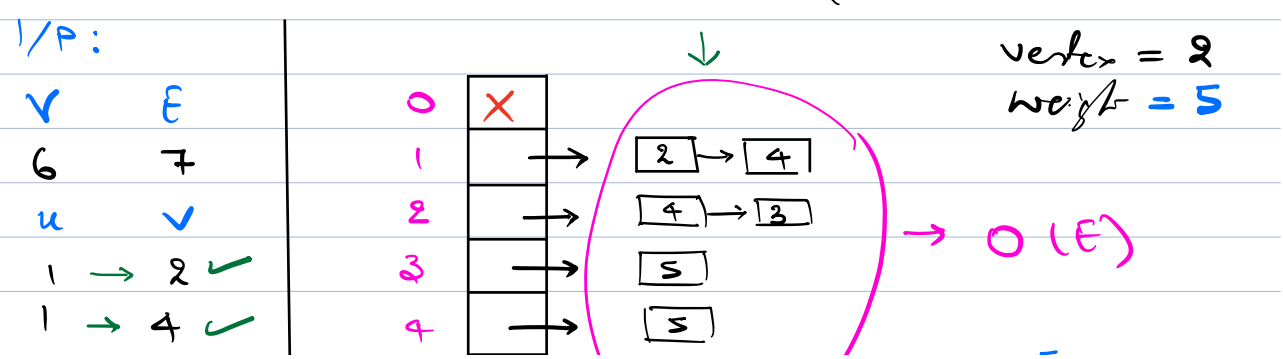
1) Space (Sparse matrix)
↓
majority are 0

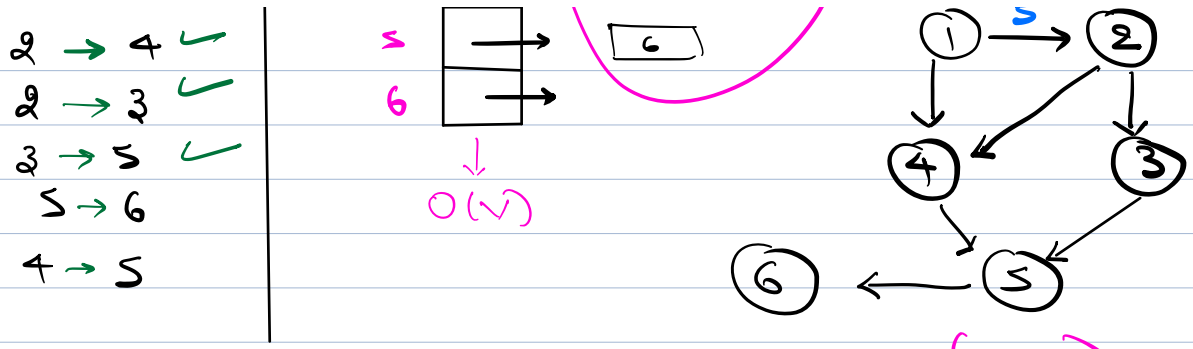
2) Add a new Node

---

2) Adjacency List (LL of Edges)
[V, E]

Array / List of size V+1 (Linked List)





S.C. = O(V+E)

LL ← Node {

int vertex;
int weight; (Weighted graph)

Node next;

}

Searching for an edge = O(V)/O(E)

Why set is not preferred?

Google Cons of using set if any??

BFS

1) BFS
2) DFS
3) CC
4)