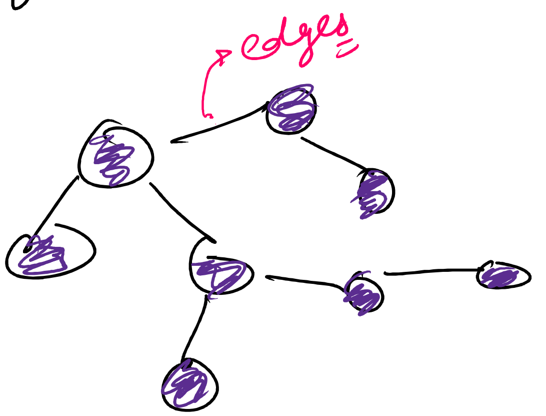
4.7 - Graphs - 1

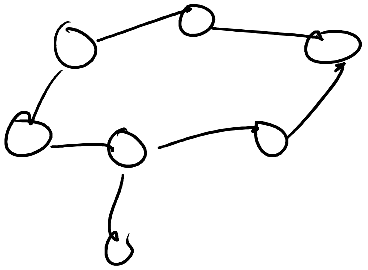
Sunday, September 7, 2025 10:50 AM

↓ Non linear Data Structures

→ Graph is a DS made up of vertices/ nodes and edges.

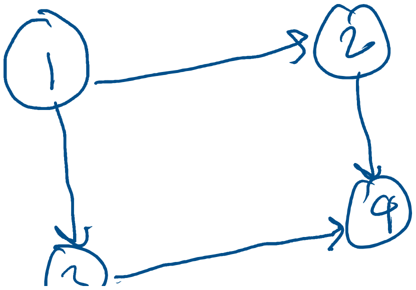


Cycle Vs Acyclic graph

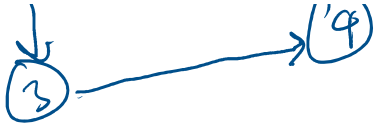


⊛ Cycle is a path which starts at a node and also ends at the same node

Directed Vs Undirected Graphs



→ this doesn't have a cycle

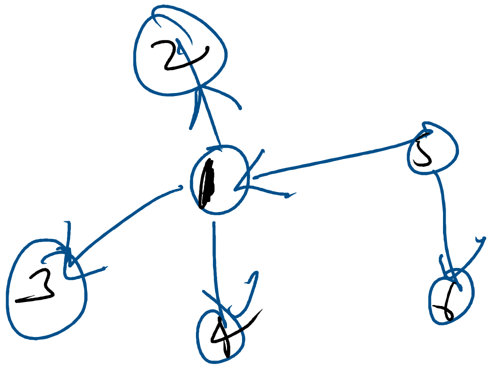


⊛ Direction is specified along every edge and we can only move in that direction.

Degree of a node

Indegree

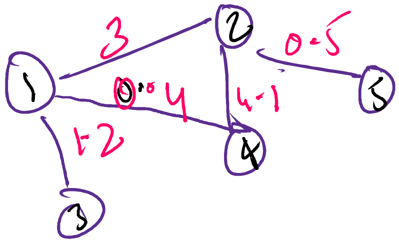
Outdegree



in deg(1) = 1

Outdeg (1) = 3

Weighted and un weighted graph



How to represent graphs

→) Adjacency matrix

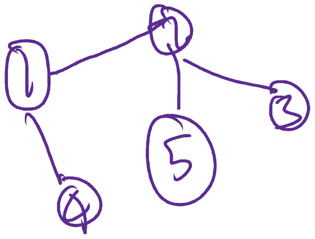
list.

→) Adjacency Matri

→) Adjacency List.

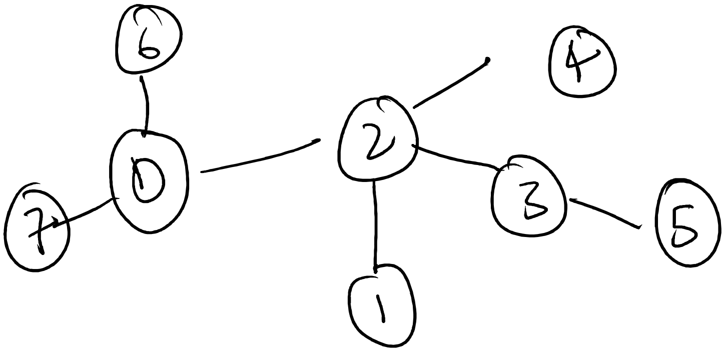
Adjacency Matrix → VxV matrix

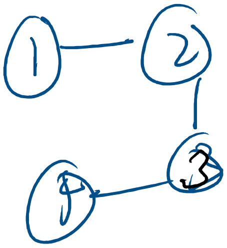
where a[i][j] represents that there is an edge from i→j



⇒ 5x5

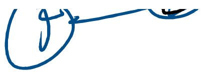
	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	1
3	0	1	0	0	0
4	1	0	0	0	0
5	0	1	0	0	0

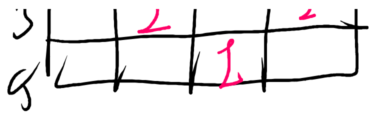




⇒

	1	2	3	4
1		1		
2	1		1	
3		1		1
4			1	





⊛ Adj Matrix wastes a lot of space for sparse graphs

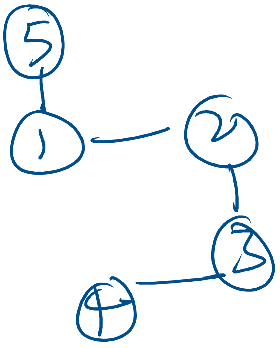
⇓

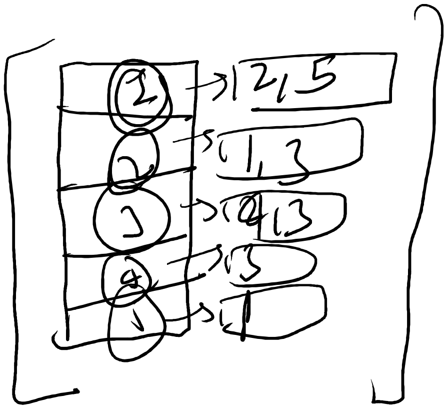
kam edges ho

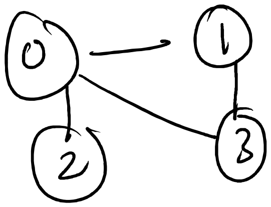
⇒ Adjacency List

⇓

Array list of array list







Traversal of graphs

1) Breadth first Search → A node traverses its immediate neighbors, the those neighbors traverse their imm nbr, continues.

$1 \to 2, 6$

$2 \to [3, 1]$

$3 \to \boxed{2, 4, 5}$

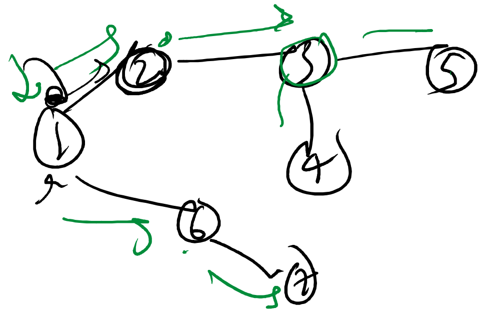
$4 \to 3$

$5 \to 3$

$6 \to 1, 7$

$7 \to 6$

⇒

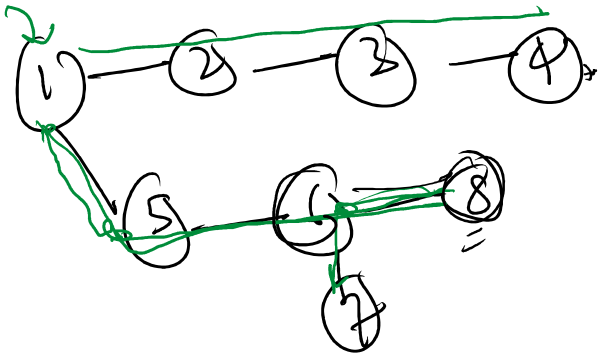


T.C = O(V+E)

⊛ BFS is implemented iteratively using queue

⇒ Depth first search

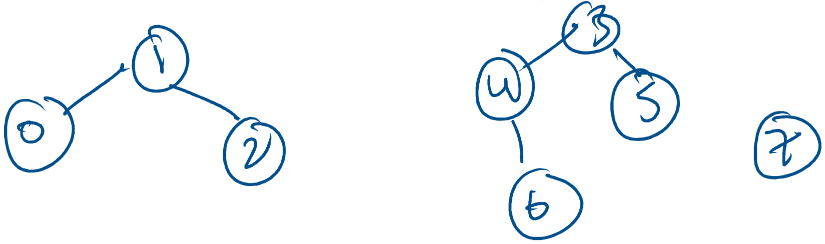
⇒ We keep traversing untill one path is not completelly traversed.



T.C = O(V+E)

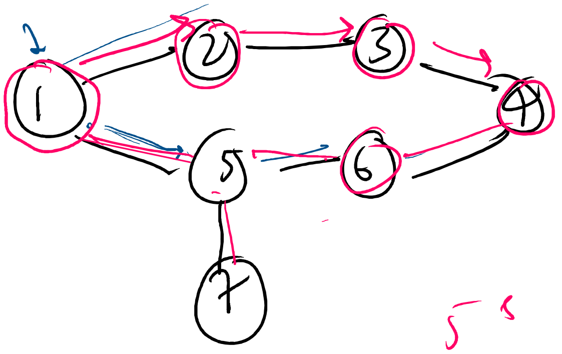
⊛ DFS is implemented recursively

Connected Vs Disconnected graphs



Disconnected graphs is which has more than one components.

Cycle in undirected graph



5 : [1, 7, 6]

cycle in directed graph

[5]]