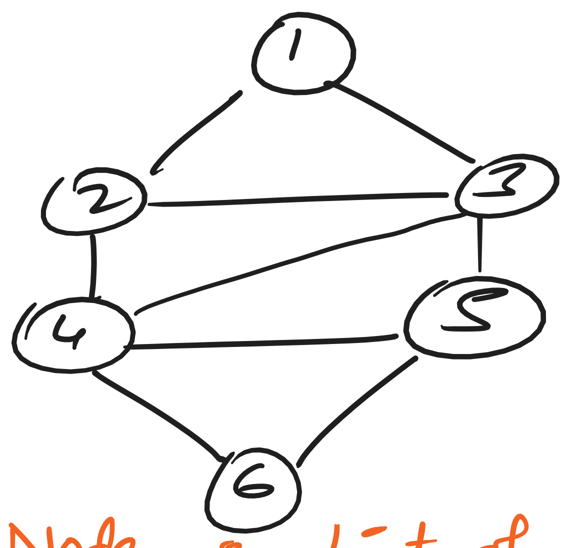
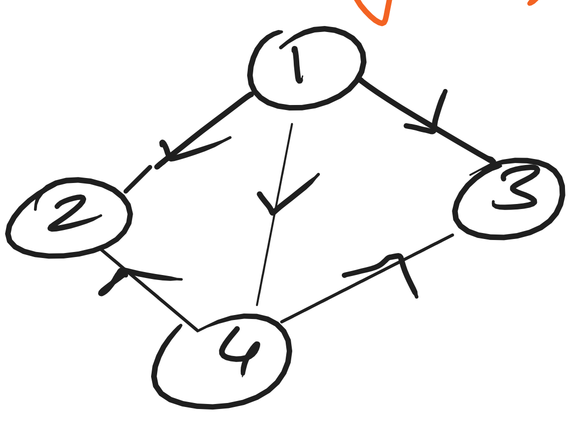
Adjacency List Representation

Undirected graph



Node : List of Neighbours
- 1 → 2, 3
- 2 → 1, 3, 4
- 3 → 1, 2, 4, 5
- 4 → 2, 3, 5, 6
- 5 → 3, 4, 6
- 6 → 4, 5

Directed graph



Node : List of Neighbours
- 1 → 2, 3, 4
- 2 →
- 3 →
- 4 → 2, 3

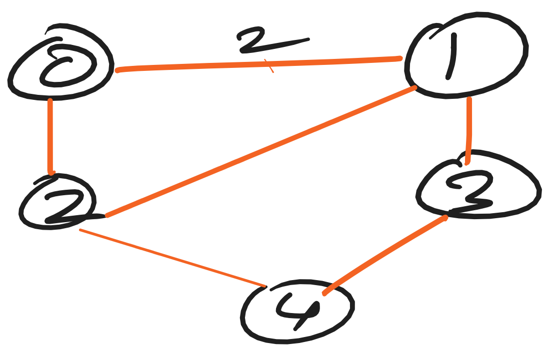
* Count the number of edges & nodes in a given " Undirected graph "

Nodes
- 0 → 1, 2
- 1 → 0, 2, 3
- 2 → 0, 1, 4
- 3 → 1, 4
- 4 → 2, 3

12 edges / 2 = 6

edges = 6
nodes = 5

scratch



* Convert a given adjacency matrix of a graph to an adjacency list.

I/P

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

Matrix (0 — 4)

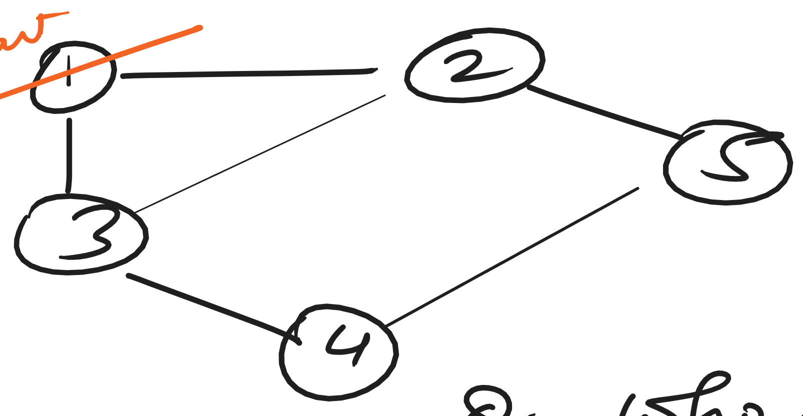
O/P

Adj List : →

Node : List of Neighbours
- 0 → 1, 2
- 1 → 0, 2, 3
- 2 → 0, 1, 4
- 3 → 1, 4
- 4 → 2, 3

Graph Traversals : → ***

1. BFS ↳ queue
2. DFS ↳ array



BFS Traversal

O/P :→ 1 2 3 5 4
(levels: 1 → 0; 2 3 → 1; 5 4 → 2)

Q. Who are your neighbours?

Adj List:
- 1 → 2, 3
- 2 → 1, 3, 5
- 3 → 1, 2, 4
- 4 → 3, 5
- 5 → 2, 4

1. Adj list
2. visited array
3. Queue

Visited Array:

1	~~F~~ T
2	~~F~~ T
3	~~F~~ T
4	~~F~~ T
5	~~F~~ T

(empty) Queue