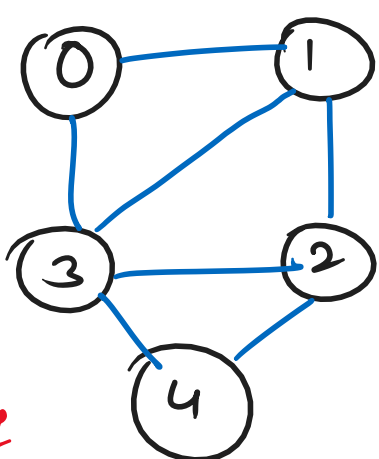


## \* Graph Interview Questions : →

① Given an undirected graph, count the number of edges & the number of nodes of the given graph : →

O/p for the given graph:

no of nodes = 5  
no of edges = 7



[Adj List]

{ 0 : 1, 3 }, { 1 : 0, 2, 3 },  
{ 2 : 1, 3, 4 }, { 3 : 0, 1, 2, 4 },  
{ 4 : 2, 3 }

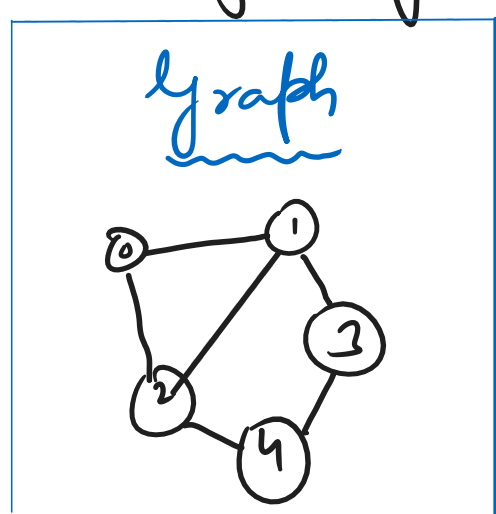
IBM = May 2025  
Cognizant = July 2025  
Wipro = June 2025

## \* HCL / IBM / Infosys / Cognizant : →

Convert a given Adjacency Matrix to List

	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 \*

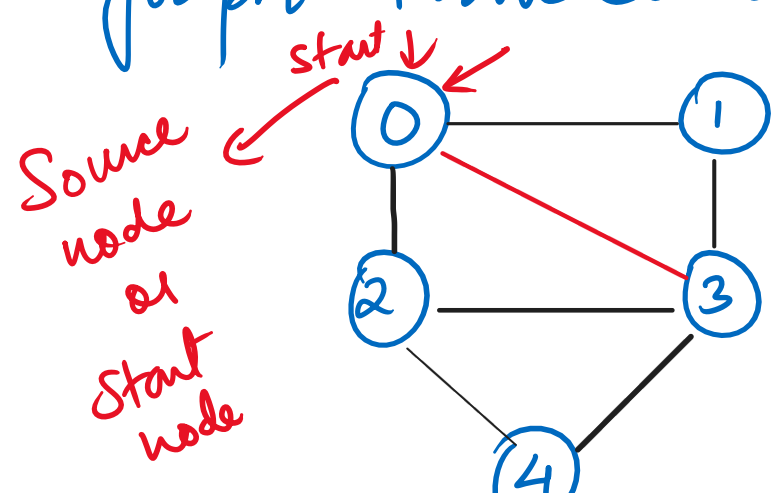


List

Node : Neighbours  
0 : (1, 2)  
1 : (0, 2, 3)  
2 : (0, 1, 4)  
3 : (1, 4)  
4 : (2, 3)

for if (adjMat[u][v] == 1) u → v  
adjList.get(u).add(v)

## Graph Traversals : → \* BFS Traversal : →



The can have multiple o/p's.  
(Level Order Traversal)

O/p: 0, 1, 2, 3, 4  
⇒ Who are your neighbours?

Prerequisites:

Node  
→ 0  
→ 1  
→ 2  
→ 3  
→ 4

Adj List:  
Neighbours:  
0 : 1, 2, 3  
1 : 0, 2  
2 : 0, 3, 4  
3 : 0, 1, 2, 4  
4 : 2, 3

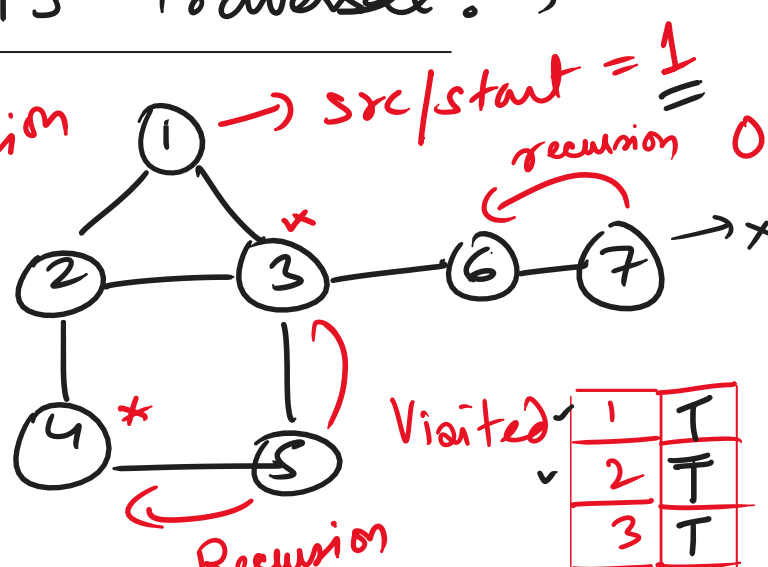
Visited

	F	T
0	F	T
1	F	T
2	F	T
3	F	T
4	F	T

(empty)  
(FIFO)  
queue

## Graph DFS Traversal : →

DFS → Queue  
DFS → Recursion



Adj List:

node 1 → (2, 3)  
2 → 1, 3, 4  
3 → 1, 2, 6  
4 → 2, 5  
5 → 3, 4  
6 → 3, 7  
7 → 6

	T
1	T
2	T
3	T
4	T
5	T
6	T
7	T

dfs(1)  
dfs(2) → X  
dfs(3)  
dfs(4)  
dfs(5)  
dfs(6)  
dfs(7)

LCP → Longest Common Prefix

\* prefix  
\* Tail

code  
coder  
coding  
codebase  
cod

(Trie)