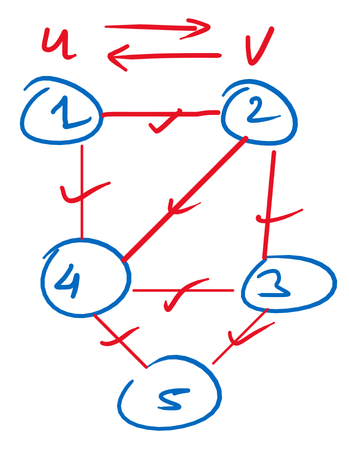
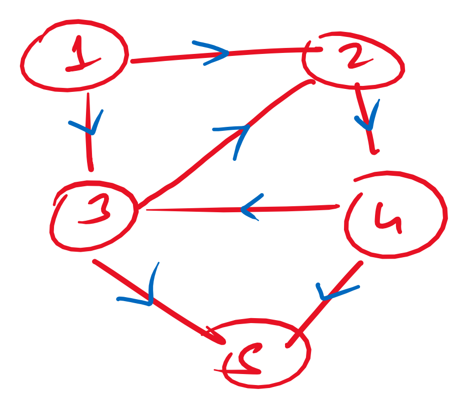
Graph Representation Using Adjacency List:



Adjacency List

Nodes	List of Neighbours
1	2, 4
2	1, 3, 4
3	2, 4, 5
4	1, 2, 3, 5
5	3, 4

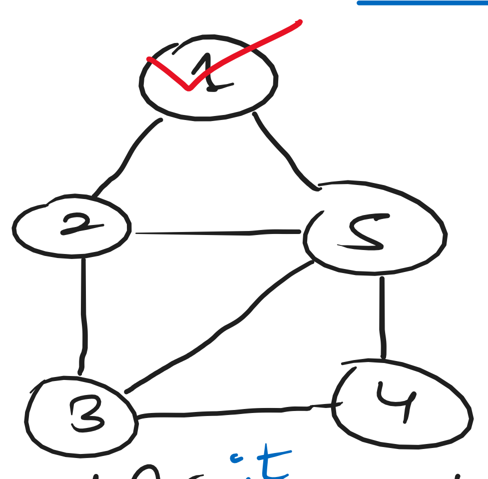


Adj List:

Node	Neighbours
1	2, 3
2	4
3	2, 5
4	3, 5
5	

*** Graph Traversal Techniques: →

① Level Order Traversal or BFS



Approach?

Starting Node = 1

o/p: 1, 2, 5, 3, 4

o/p: 5, 1, 2, 3, 4

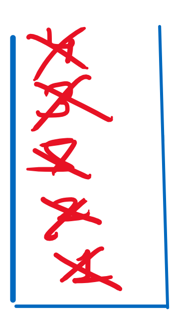
Question? who are your neighbors? Adj List

Prerequisites:

Visited

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

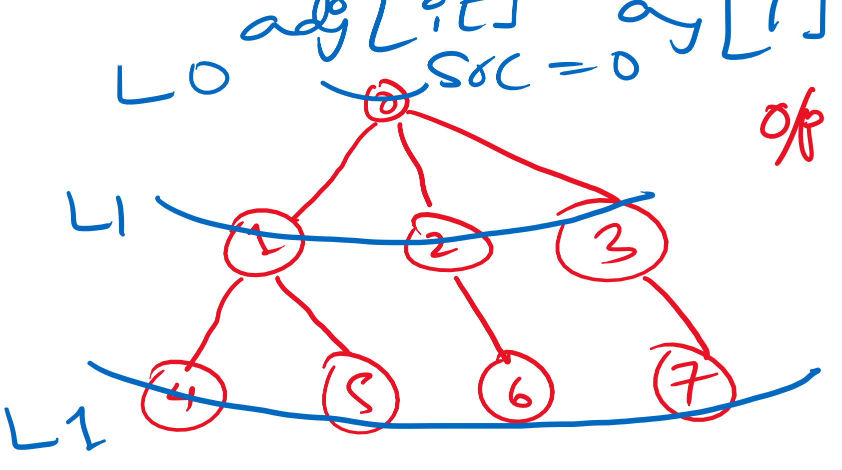
Queue



Adjacency List:

1 → 2, 5
2 → 1, 3, 5
3 → ~~2~~, ~~4~~, ~~5~~
4 → 3, 5
5 → ~~1~~, ~~2~~, ~~3~~, 4

adj[it] adj[1] = 2, 5



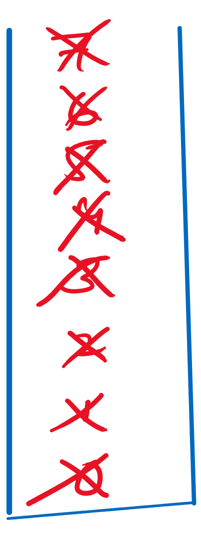
o/p → 0, 1, 2, 3, 4, 5, 6, 7

Adj List:

✓0 → 1, 2, 3
✓1 → 0, 4, 5
✓2 → 0, 6
✓3 → 0, 7
✓4 → 1
5 → 1
6 → 2
7 → 3

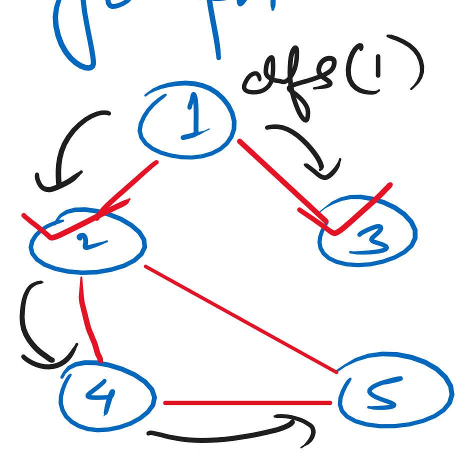
Visited

0 → ~~F~~ T
1 → ~~F~~ T
2 → ~~F~~ T
3 → ~~F~~ T
4 → ~~F~~ T
5 → ~~F~~ T
6 → ~~F~~ T
7 → ~~F~~ T



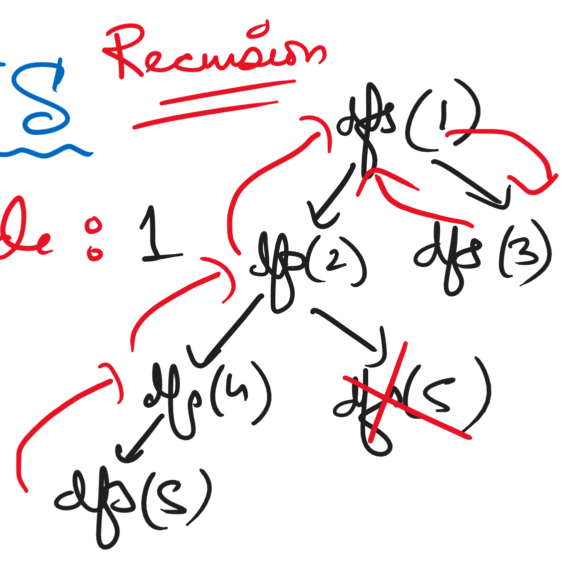
Graph Traversal: DFS

Recursion



Source Node: 1

①



Idea:

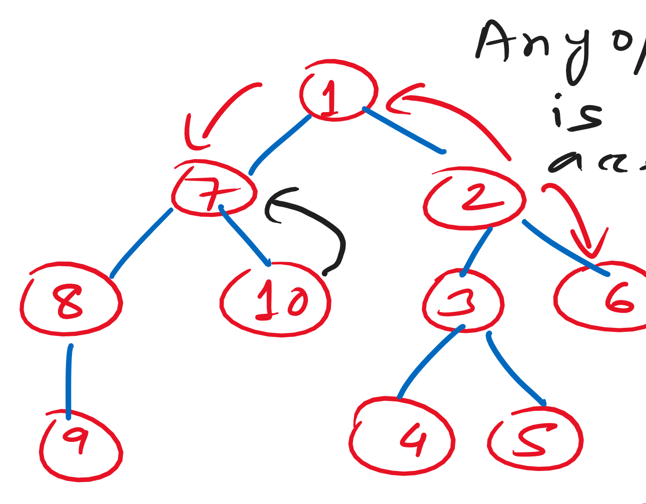
1, 2, 4, 5, 3

Adjacency List:→

→ 1 → 2, 3
2 → 1, 4, 5
3 → 1
4 → 2, 5
5 → 2, 4

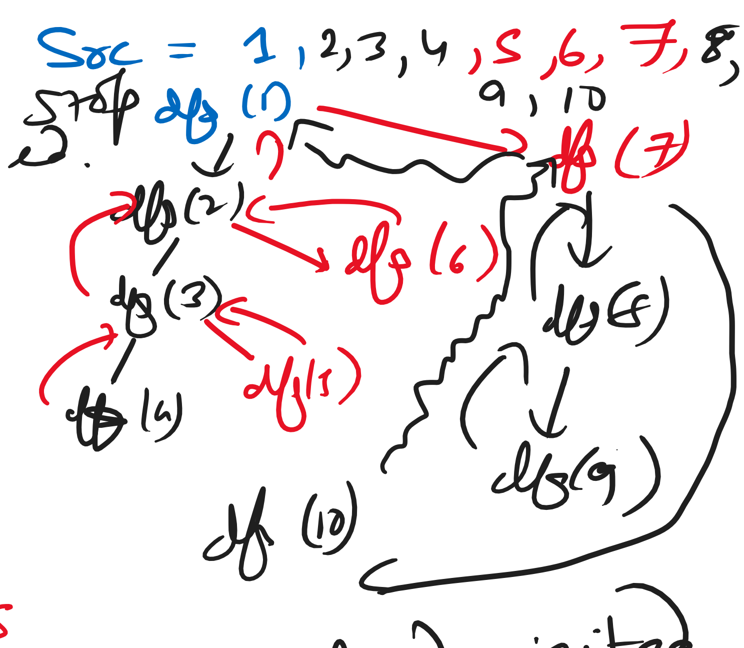
Visited Array/Vector

Nodes→	1	2	3	4	5
	1	1	1	1	1



Any o/p is accepted.

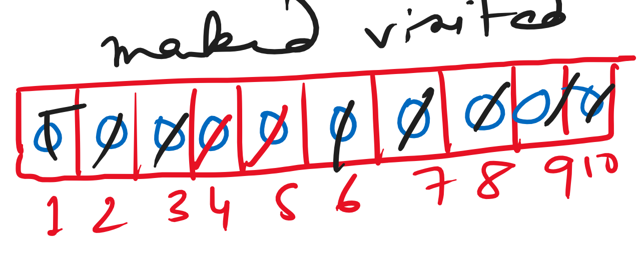
Src = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10



Adj List:→

1 → 2, 7
2 → 1, 3, 6
3 → 2, 4, 5
4 → 3
5 → 3
6 → 2
7 → 1, 8, 10
8 → 7, 9
9 → 8
10 → 7

marked visited



JDBC
Java → SQL
Generics
Stream API

Graphs → Adj Mat, Adj List
Types, Theory,
Representation
& Traversals:
BFS & DFS

Phase I

Post Lunch: →

Introduction to Dynamic Programming

- * Recursion
- * Memoisation
- * Tabulation
- * Space Optimisation
- * Problems