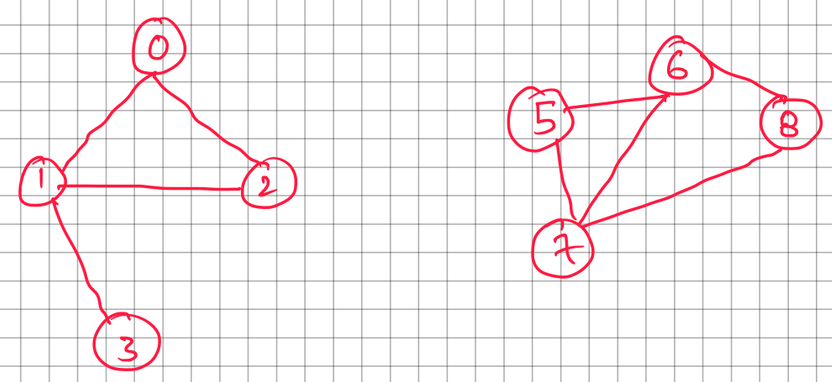
Path Finding

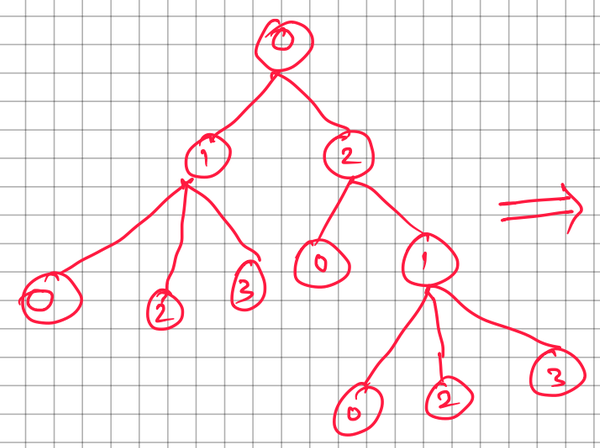
#) Un Directed Graph



#) Memory Representation

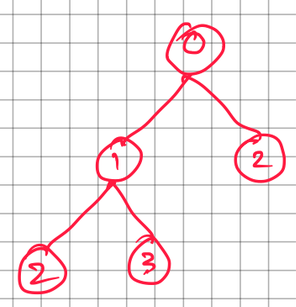
	0	1	2	3	4	5	6	7	8
0	0	1	1	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	1	1	0
6	0	0	0	0	0	1	0	1	1
7	0	0	0	0	0	1	1	0	1
8	0	0	0	0	0	0	1	1	0

#) Search Path(0, 3) = ?



⟹ Note:-) Problem :-) We are Re-visiting the Node's again & again.
So, We have to Keep track of Re-visiting node's.
↳ Solution:-) Visiting Array.

#) Without Re-visiting Nodes



⟹ "Reduced Function Call's"

⟹ (2) Possible Path's

- Path 1 :-) 0 - 1 - 3
- Path 2 :-) 0 - 2 - 1 - 3

#) Algorithm Design

(Source, Destination)

↳ if (S == D) Return T

→ Else if: Not found (Explore Intermediate Node's)

Decompose: (Source(i), Destination)

(S1, D) (S2, D) . . . . . (Si, D)

. . . . .

Recompose: If found: End (Return True)

If Not: End (Return false)