Topo - Sort :- (If there is edge u→u ; u appear before v)

① Only applicable in DAG

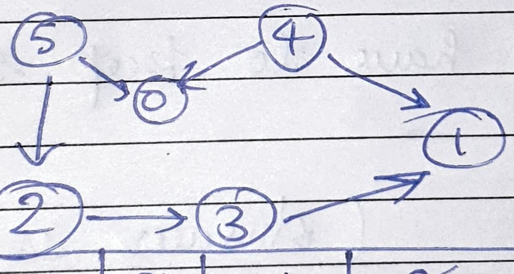
② 2 method → a) BFS (kahn's algo)
→ b) DFS

a) BFS :- make a inorder array that stores in- degree of ith node.

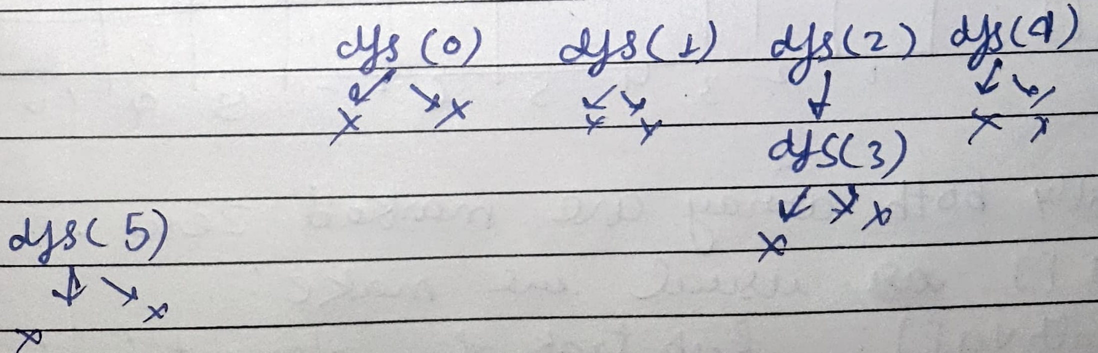
Initially; Now push all nodes into queue which are having in-degree = 0. and now apply BFS algo. and also make a vis array..

(b) DFS :- initialize an empty stack.
(Intution :- u→v is an edge suppose ; before pushing to 'u' to stack first explore the v (call for dfs(v)) and then push u.

Ex →



vis[]:	~~0~~, 1	~~0~~, 1	~~0~~, 1	~~0~~, 1	~~0~~, 1	~~0~~, 1
	0	1	2	3	4	5



5
4
2
3
1
0

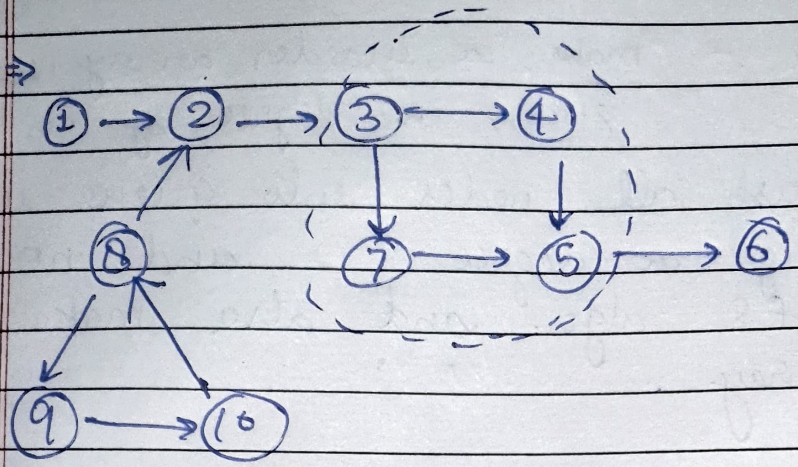
Stack

now poped element from stack are stored in container. This container has one of Topo sort,

① Detect cycle using BFS in DG :-
Apply Kahn algo & observe.

② Detect cycle in DG using DFS :-

Ex ⇒



Why can't we simply use dfs as in undirected G

~~Make~~ we can't bcz in above example ; The highlited part doesn't contain cycle but using simple dfs it will tell ~~dfs~~ cycle is present.

So ; we have to keep something else.

But what ? ?

(Answer is in next page).

	1	2	3	4	5	6	7	8	9	10
Vis [] :										
PathVis [] :										

Intially both array are marked zero.

Vis [] as usual we make.

PathVis [] : keep trak of element/nodes that are part of current stack space of dfs call.