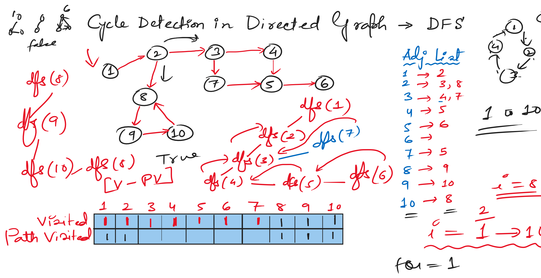
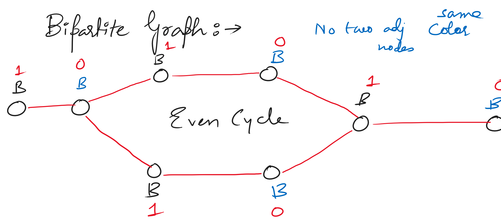
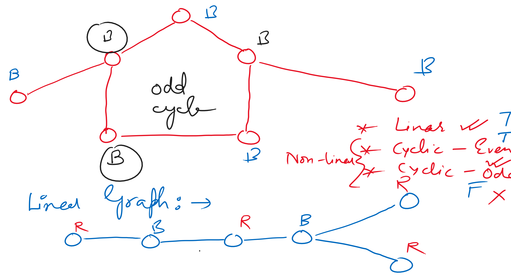
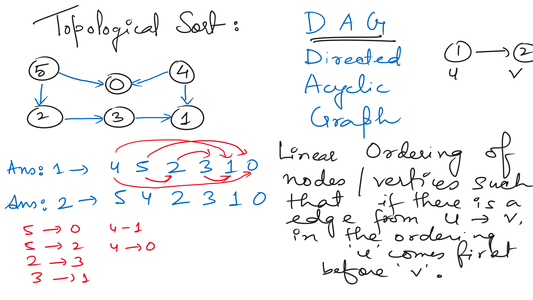
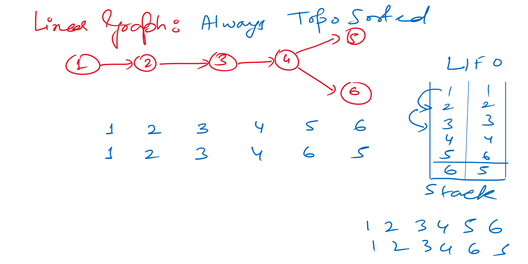
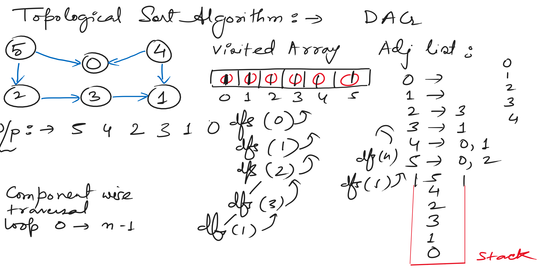

Cycle Detection in Directed Graph → DFS
Adj List
dfs(5)
dfs(9)
dfs(10)
dfs(6)
True
[V - PV]
dfs(1)
dfs(2)
dfs(3)
dfs(4)
dfs(5)
dfs(6)
dfs(7)
Visited
Path Visited
for i = 1

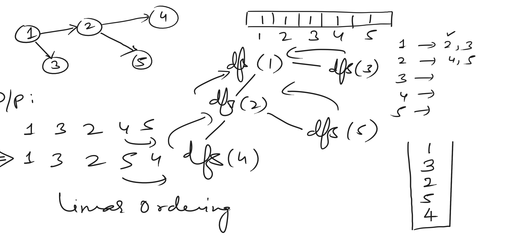

Bipartite Graph: →
No two adj nodes same Color
Even Cycle

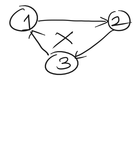

odd cycle
Linear ✓ T
Cyclic - Even ✗ T
Non-linear
Cyclic - Odd ✓ F
Linear Graph: →


Topological Sort:
DAG
Directed
Acyclic
Graph
Ans: 1 → 4 5 2 3 1 0
Ans: 2 → 5 4 2 3 1 0
Linear Ordering of nodes / vertices such that if there is a edge from u → v, in the ordering 'u' comes first before 'v'.


Linear Graph: Always Topo Sorted
LIFO
Stack


Topological Sort Algorithm: → DAG
Visited Array
Adj List:
O/P: → 5 4 2 3 1 0
Component wise traversal loop 0 → n-1
Stack


O/P:
1 3 2 4 5
⇒ 1 3 2 5 4
Linear Ordering


DAG →
1 → 2 → 3
1 → 2 → 3