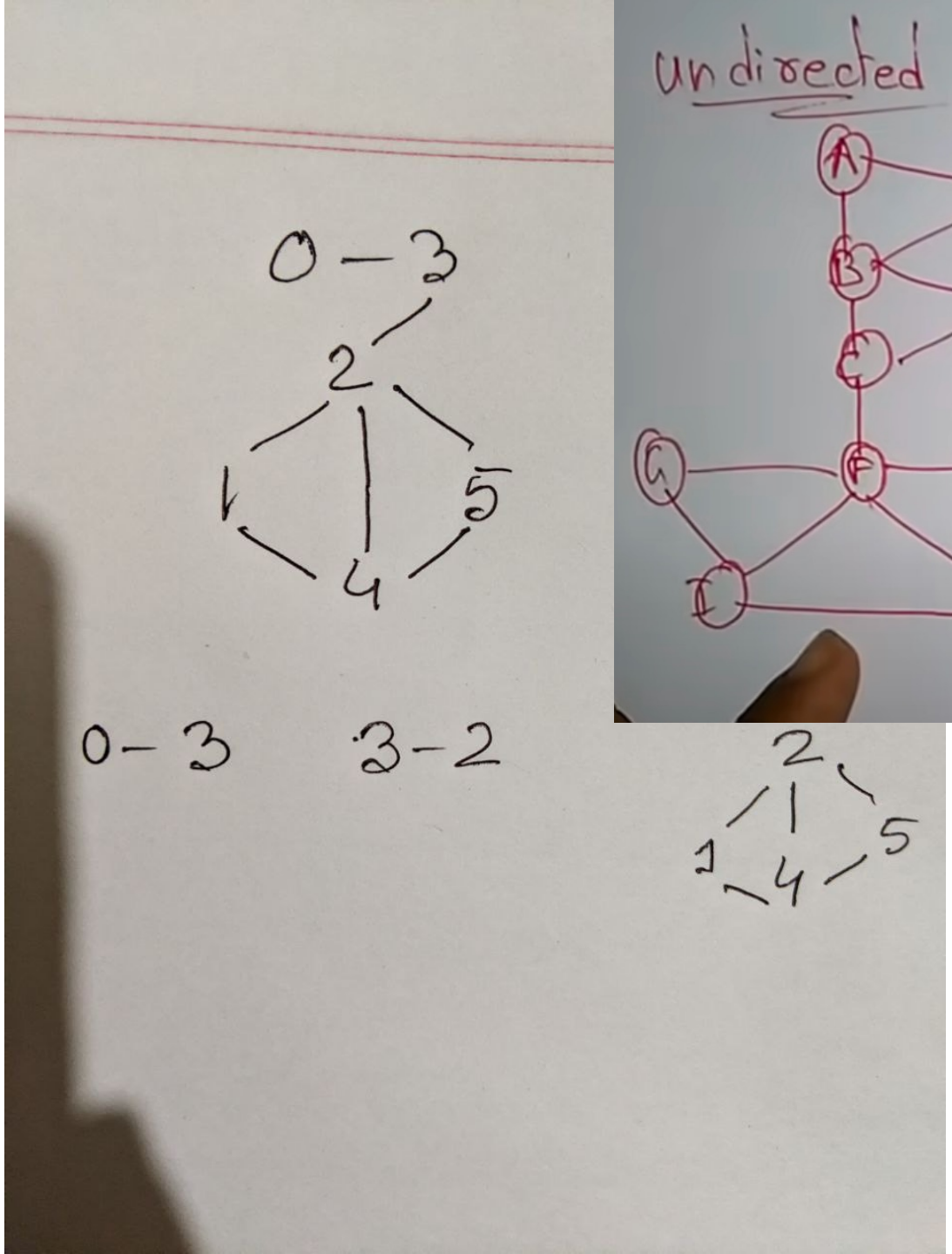
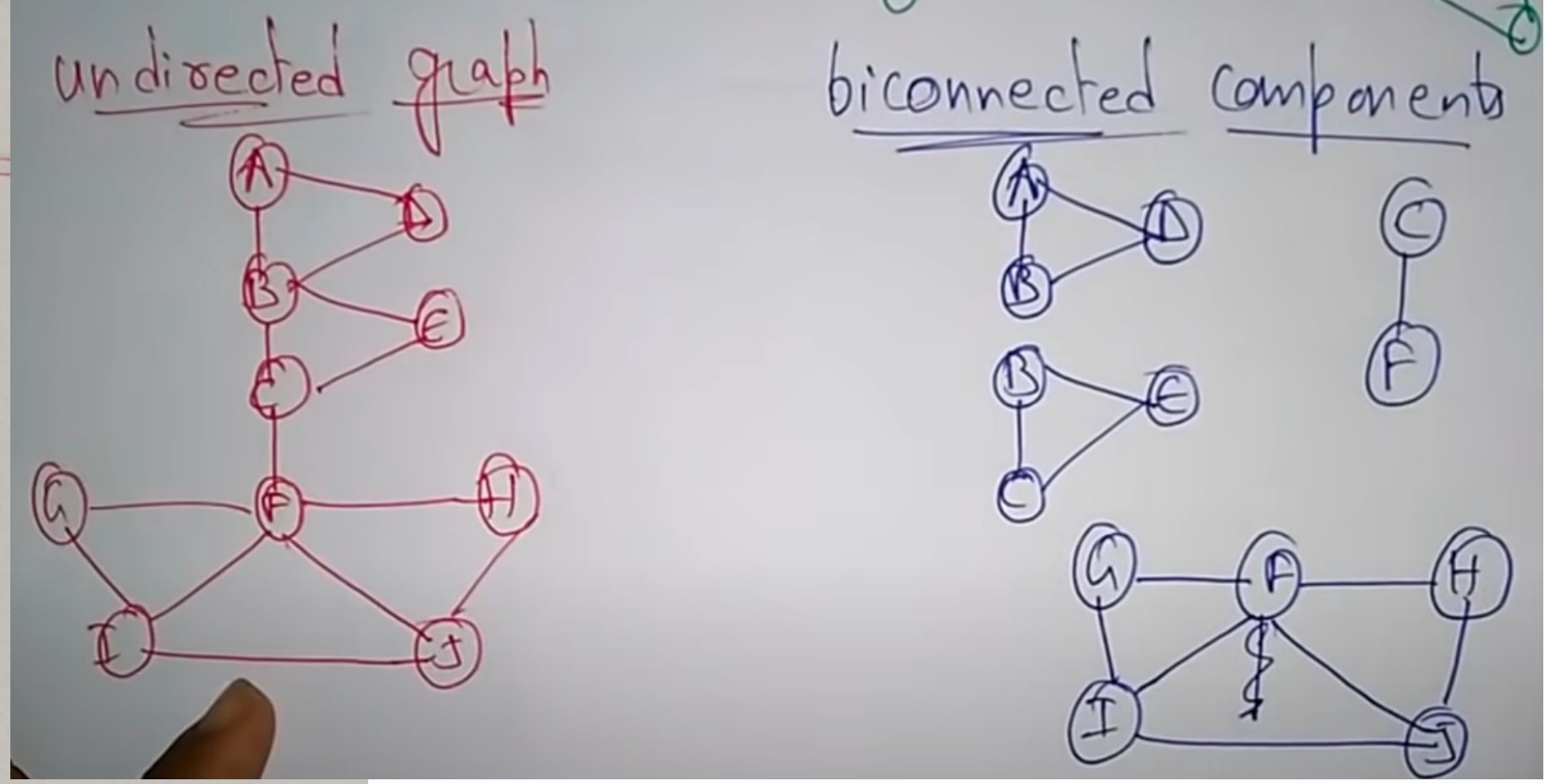
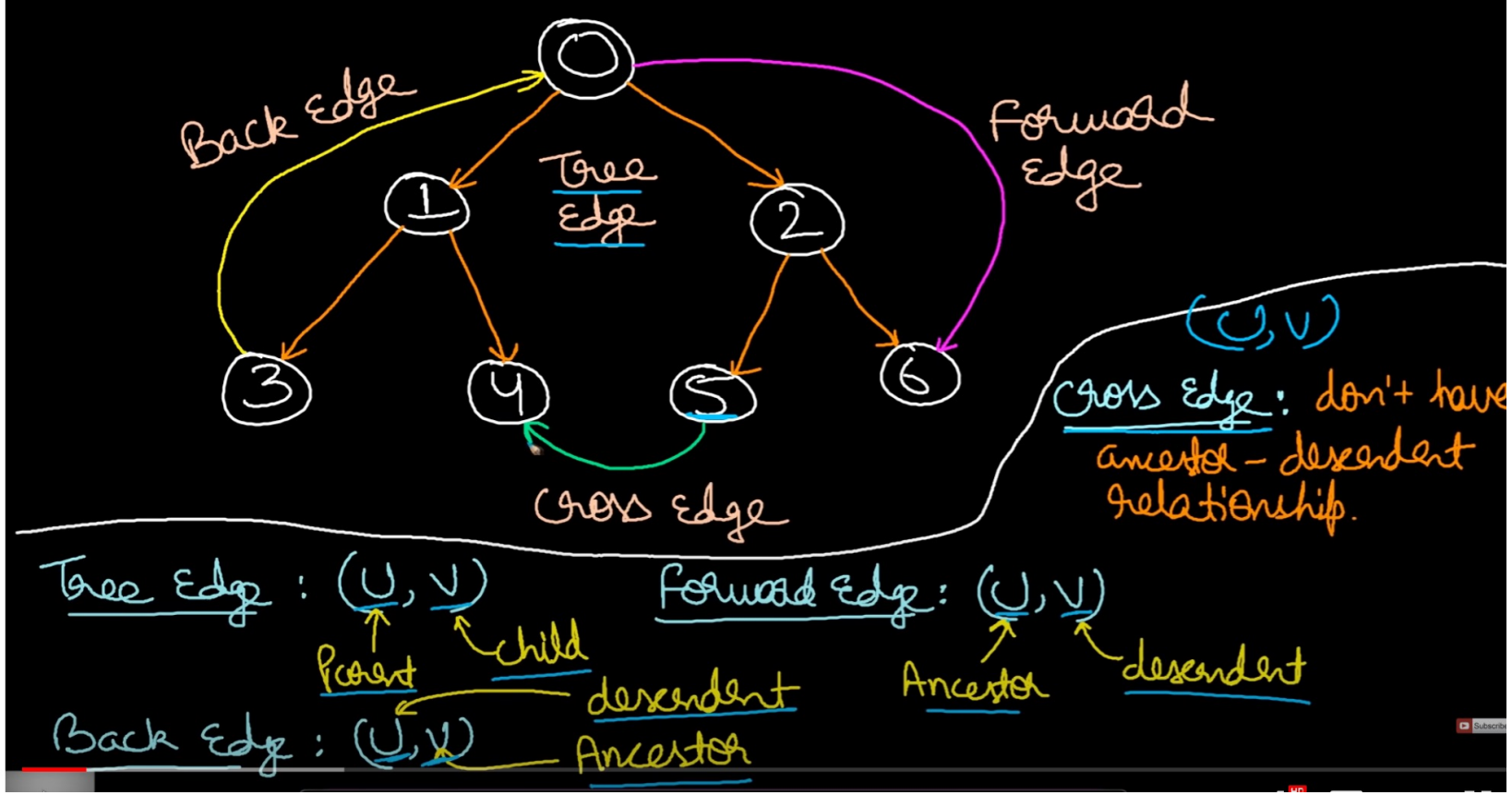
Bcc:


0 - 3
2
1
5
4
0 - 3
3 - 2
2
1
5
4


undirected graph
A
D
B
E
C
G
F
H
I
J

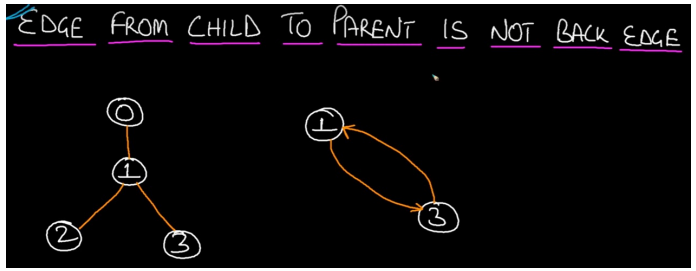
biconnected components
A
D
B
C
F
B
E
C
G
F
H
I
J

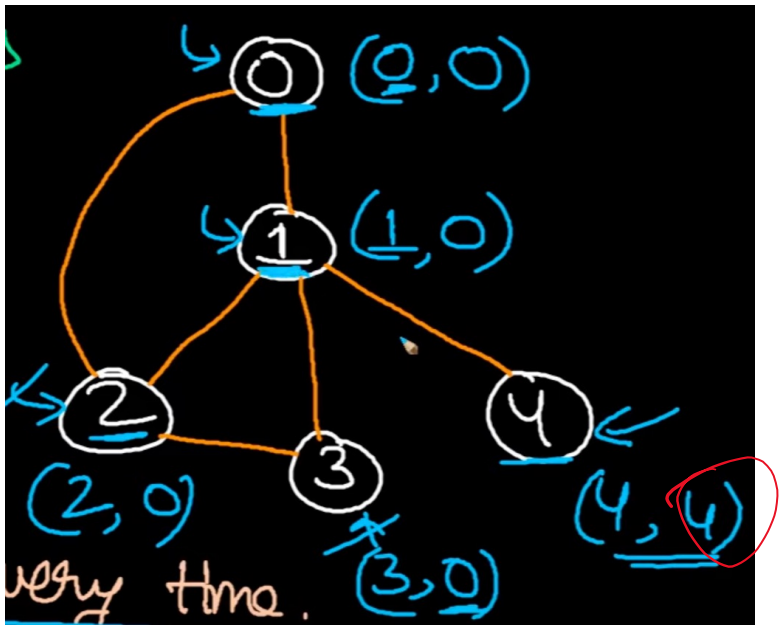


https://youtu.be/ZeDNSeilf-Y?t=97

Low value is also discovery time .curr node theke accessible shokol node er moddhe je node er disc time min

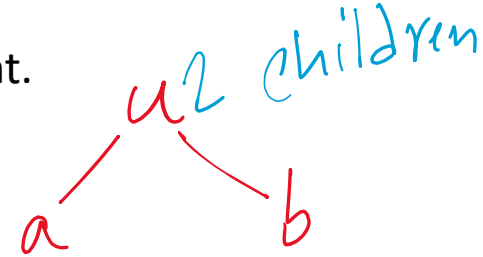
For directed graph:

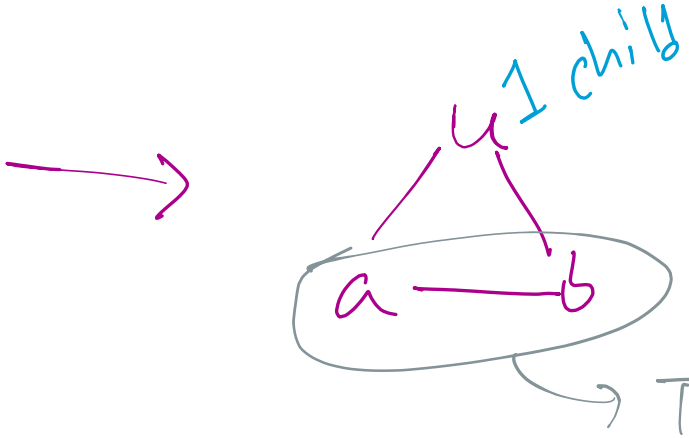




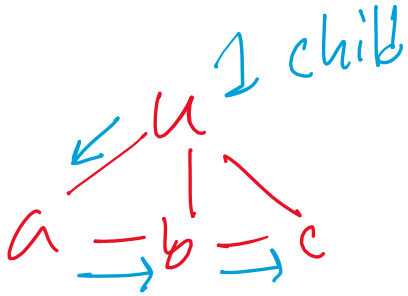
Case 1:

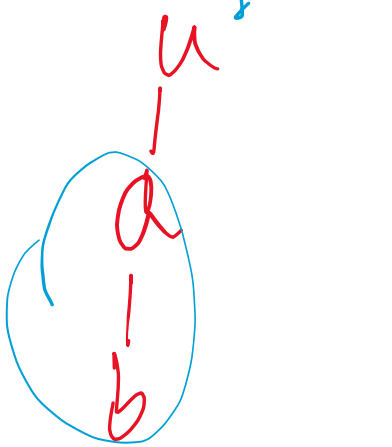
If u is the root node of DFS tree and has at least two children(a,b) then it is a articulation point.



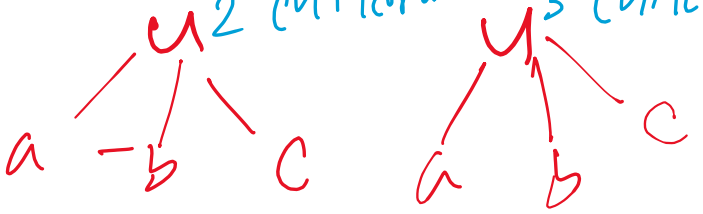


This is the only subgraph of u here.



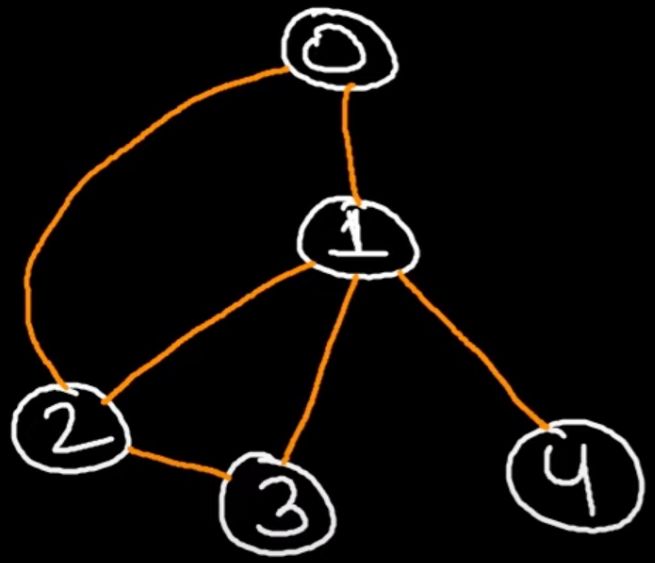


According to DFS, after visiting a, b will be visited. b will not visit as the child of a. a—b will be visited as the subgraph of u ∴ u has one child in the DFS tree.

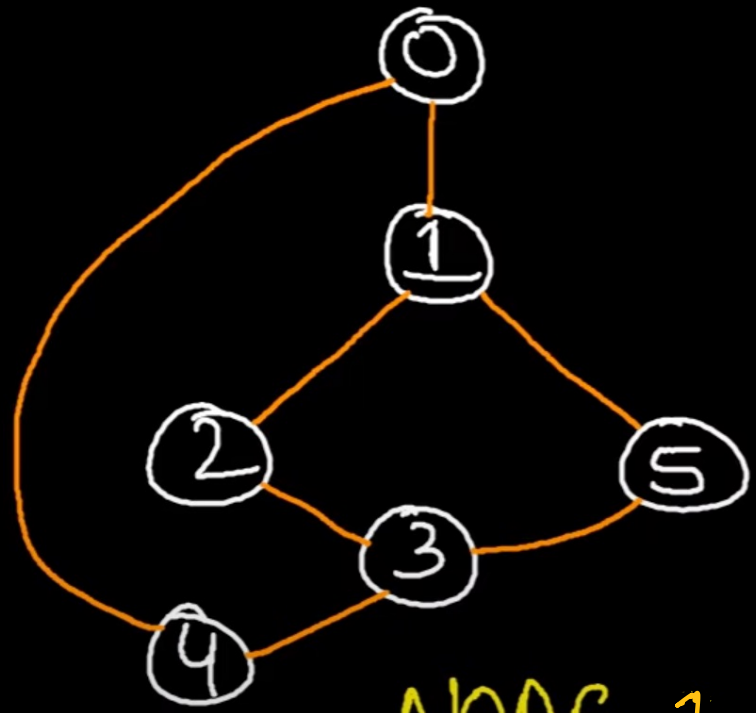


Case 2

2) Node U is not root of DFS tree and it has a child v such that no vertex in subtree rooted with v has a back edge to one of the ancestors of U.



Node-1 is AP

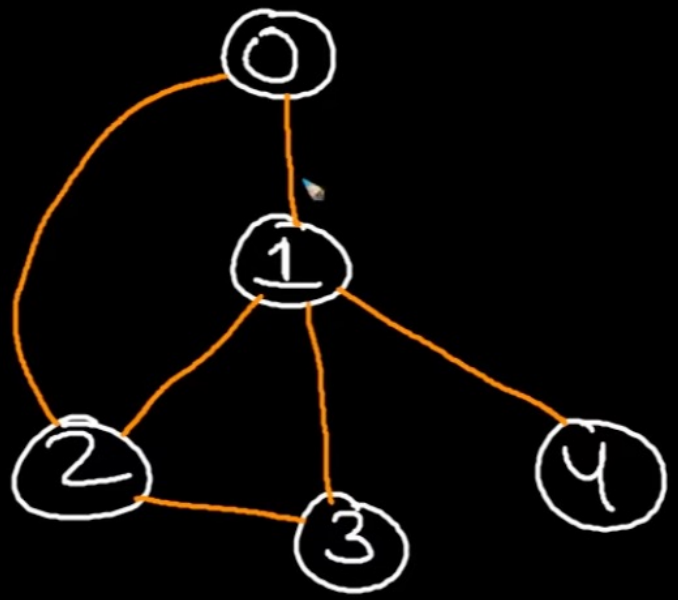


NODE-1 is NOT AP

https://youtu.be/64KK9K4RpKE?t=784

HOW TO DETECT AP FOR CASE-2

1. we need to find the order of vertices from earliest to latest to detect back-edges.



∴ we use timeStamp to mark Nodes with increasing value.

∴ we can do this by assigning discovery time.

2. we need to maintain the earliest possible Node accesible for a given Node which will indicate if we have any back edge.

For that we will assign Low value to each Node.