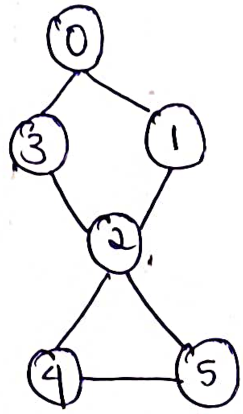
3 Cut-vertices / Find Articulation point :

Articulation point : Removal of the vertex & associated edges will disconnect the graph.

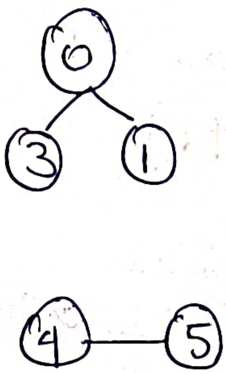
Eg



Here 2 is articulation point.

∵ If you remove 2

→



Condition

Consider (u, v)

u → parent

v → child

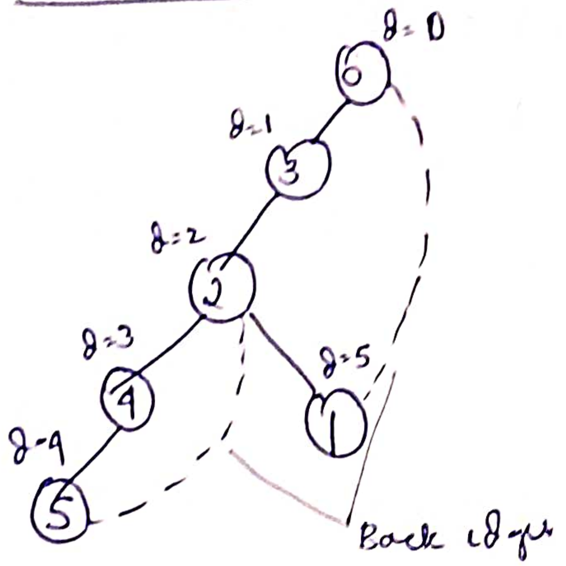
If $L[v] \geq D[u] \Rightarrow u$ is articulation point

↓

This condition holds good for all nodes, except root node.

Tracing

Apply DFS



vertex	0	1	2	3	4	5
d	0	5	2	1	3	4
L	0	0	0	0	2	2

d → discovery time. i.e; the order in which you visit the node in DFS

L → parent node through back edge.

Note:- If you start d=0, then you can take L = parent node directly.
If you start as d=1, then you should take L = [parent node + 1]

Edge (u, v)	Formula $L[v] \geq d[u]$
$(\overset{u}{3}, \overset{v}{2})$	$L[2] \geq d[3]$ $\boxed{0 \geq 1}$ ⇒ False ⇒ 3 is not AP.
$(\overset{u}{2}, \overset{v}{4})$	$L[4] \geq d[2]$ $\boxed{2 \geq 2}$ ⇒ True ⇒ 2 is AP

Time complexity:

Since finding AP involves finding DFS also. Its complexity is same as that of DFS.

$O(n)$

where $n \rightarrow$ no of vertices

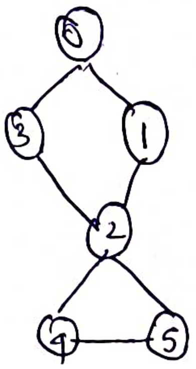
We can also say,

we scan the array L[], D[], once. It has n elements

$\therefore$ Time complexity = $O(n)$

Program Tracing:

adjacency list:



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

output

Articulation point : 2