③ Implement an algorithm for finding cut vertices using depth-first search (DFS).

A vertex is said to be a cut vertex or articulation point in a graph if removal of the vertex and associated edges disconnects the graph. So the removal of cut vertex increases the number of connected components in a graph.

cut vertices are sometimes called Articulation point.

How to find all cut vertices in a given graph?

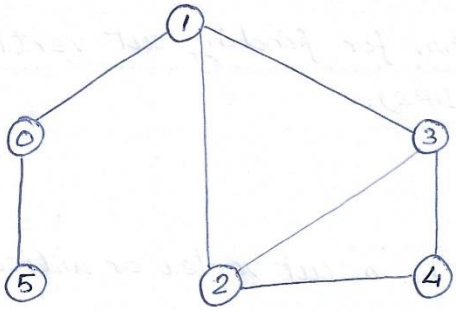
Since we want to use DFS (Depth First search). In DFS, we follow vertices in tree form called DFS Tree. In DFS Tree, a vertex u is parent of another vertex v, if v is discovered by u.

In DFS Tree, a vertex u is cut-vertex if one of the following two conditions is true.

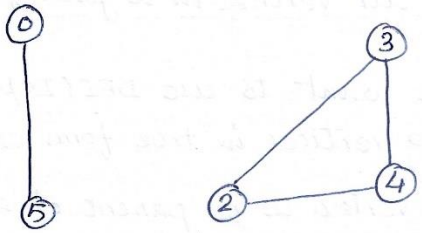
(i) u is root of DFS tree and it has atleast two children.

(ii) 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 (in DFS tree) of u.

Example:

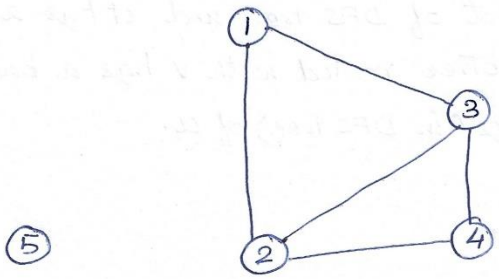


If in the above graph, vertex-1 and all the edges associated with it, i.e. the edges 1-0, 1-2 & 1-3 are removed, there will be no path to reach any of the vertices 2, 3 or 4 from the vertices 0 and 5. That means the graph will split in to two separate components as below:



so (1) is a cut vertex

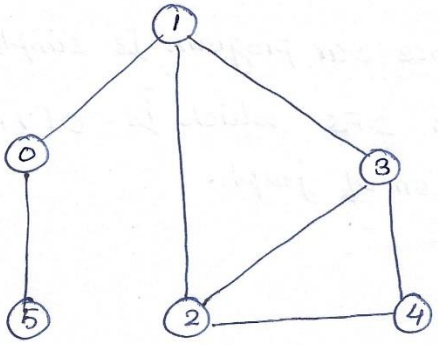
Likewise, removing the vertex-0 will disconnect the vertex-5 from all other vertices



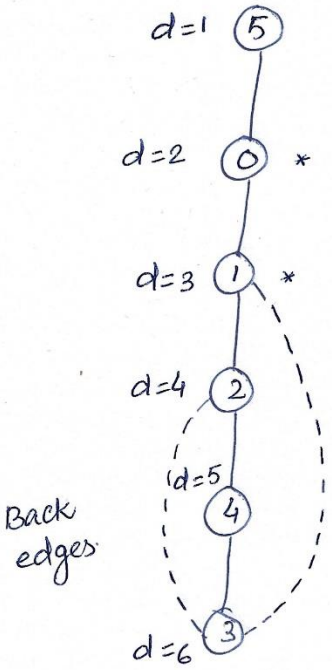
So vertex (0) is cut vertex

How to Find cut vertices using DFS (Depth First search):

Given graph



We can find DFS minimum spanning Tree:



here d is DFS search number or discovery time.

Nodes	0	1	2	3	4	5
Discovery time	2 *	3 ▲	4	6	5	1
Lowest discovery time by taking one back edge	3	3 *	3 ▲	3	3	3

A node u is cut vertex if v vertex is child of vertex u then
Lowest discovery time (v) ≥ discovery time (u)

so from above table we can find that

vertex 0 and vertex - 1 are cut vertices.

Time complexity:

Since our program is simple DFS. so time complexity is same as DFS which is $O(V+E)$ for adjacency list representation of graph.