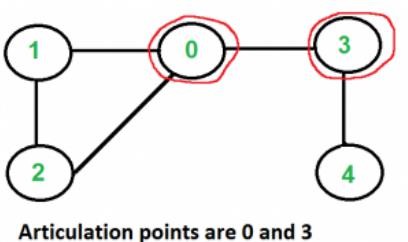
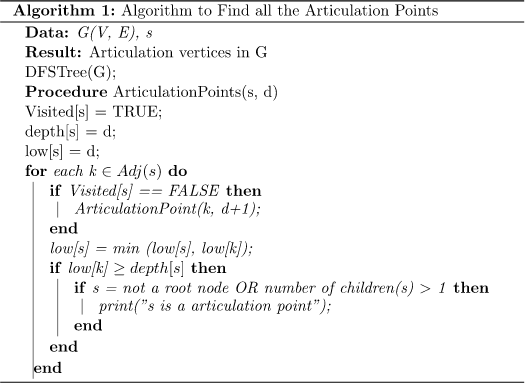
# Finding Articulation Point in a Graph:

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

* + Articulation points are sometimes called cut vertices. The main aim here is to find out all the articulations points in a graph.
  + **Example:** Given an undirected graph G, then find all the articulation points in the graph.

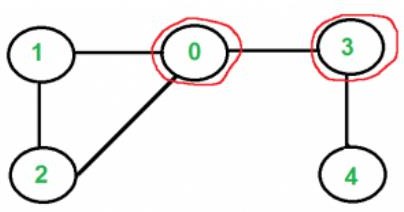
**Input:** { Below Graph}



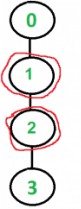
**Output = 0,3 Algorithm:**

* + This is a **DFS** based algorithm to find all the articulation points in a graph. Given a graph, **the algorithm first constructs a DFS tree.**
  + Initially, the algorithm chooses any random vertex to start the algorithm and marks its status as visited. **The next step is to calculate the depth of the selected vertex.** The depth of each vertex is the order in which they are visited.
  + **Next, we need to calculate the lowest discovery number.** This is equal to the depth of the vertex reachable from any vertex by considering one back edge in the DFS tree. An edgeis a back edge ifis an ancestor of edgebut not part of the DFS tree. But the edgeis a part of the original graph.
  + **After calculating the depth and lowest discovery number for the first picked vertex, the algorithm then searches for its adjacent vertices.** It checks whether the adjacent vertices are already visited or not. If not, then the algorithm marks it as the current vertex and calculates its depth and lowest discovery number.

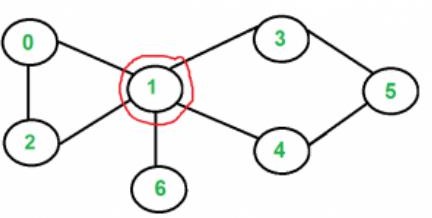
**Graph 1: (Articulation Point Graph 1:0,3)**



**Graph 2:(Articulation Point Graph 2: 1,2)**

****

**Graph 3: (Articulation Point Graph 3: 1)**

****