**Graph-Theory – Articulation Points**

1. What is an articulation point?

Ans: node whose removal disconnects the graph into two or more components.

2. What is Ancestor node?

Ans: A node other than the parent node which has been traversed prior is an ancestor node.

3. What is back-edge?

Ans: An ancestor node with 3 must-conditions:

1. It has been visited prior to this node.
2. It is still present in the stack (during DFS visit) while the current node is being visited.
3. There is an edge from the current node to the given ancestor node.

5. How to populate disc[]

Ans: disc[] = Discance of each node as we traverse DFS in the graph.

So, disc[start-node] = 0,

Increment disc[] value for each node as we detect the next node.

node 0 -> disc = 0

node 1 -> disc[0]++ = 1

node 2 -> disc[1]++ = 2

node 3 -> disc[2]++ = 3

4. How to populate min[]

Ans:

min[] -> The minimum discance needed to cover, from one node to another node.

(Generally one of the ancestor node. Note ancestor node is different from parent node).

Case 1: When nodes, u and v are immediate connected nodes.

min[u] = min(min[u], min[v]).

Case 2: When v is a backedge.

Min[u] = min(min[u], disc[v]). Where disc[v] is the disc[ ] value of backedge and min[u] is the min[ ] value of current node whose min is getting calculated.

Case 3: min[u] in first traversal.

In first traversal of a node, min[u] = disc[u]. Since, any future node is not seen yet.

5. Articulation point detection.

Ans:

1. For non-root node, If min[u] >= disc[v], then u is an articulation point. (Where u is a child, and v its parent.). Then v is an articulation point.
2. For root-node, if children[v] > 1, then root u is an articulation point.

(0, 0)

(1, 1)

(2, 2)

(3, 3)

In above graph, lets consider node 3. It’s (u, v) = (2, 2). If this node (non-root) is removed, the graph is split between 2 sub-graphs, { (1, 2) and (4)}. Hence node 3 is an articulation point. Also, you can clearly see, min[3] = 2, is greater than disc[2] = 1. Hence, for articulation point. Min[u] >= disc[v].

6. Modification of min[ ] values for each node.

Ans: The min[ ] values keeps modifying, on every iteration whenever the node is popped and the stack modified. The popped node is compared again with previous min value, and modified.

7. Back edge vs cross edge determination.

Ans: An edge which is previously traversed and is in the adjacency list of current node, can be either a back-edge or a cross-edge.

Back-edge: If the previous visited node is still in stack, else it is cross-edge.