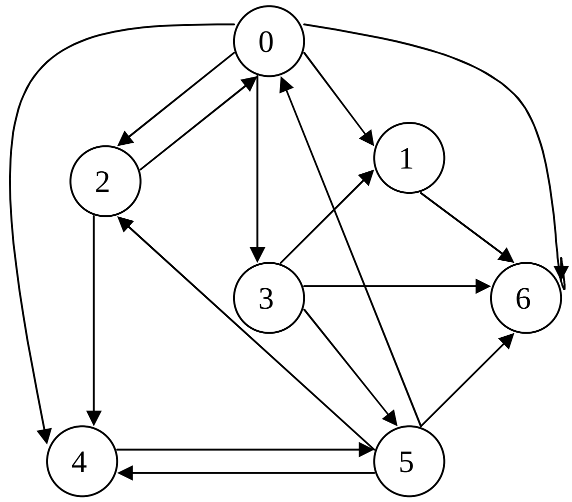
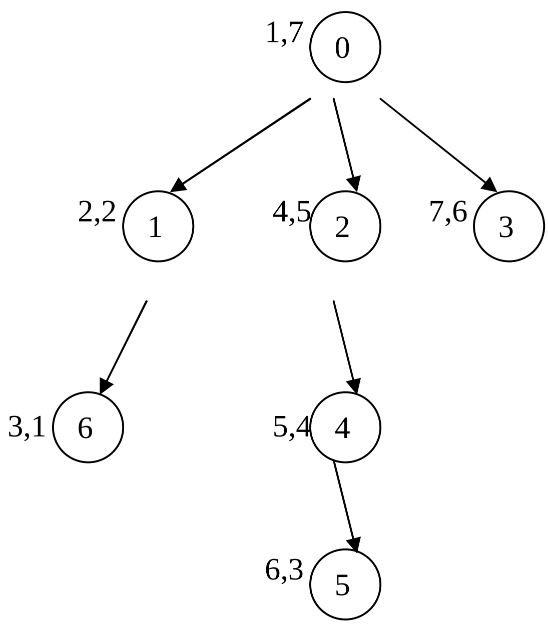
TUTORIAL-6

Example-1: Draw depth-first search (DFS) tree of the following digraph originating from vertex 0 and label the vertices with pre-order and post-order labels.



DFS Tree



Example-2: Write down all Tree arcs, Back arcs, Cross arcs and Forward arcs for the above digraph.

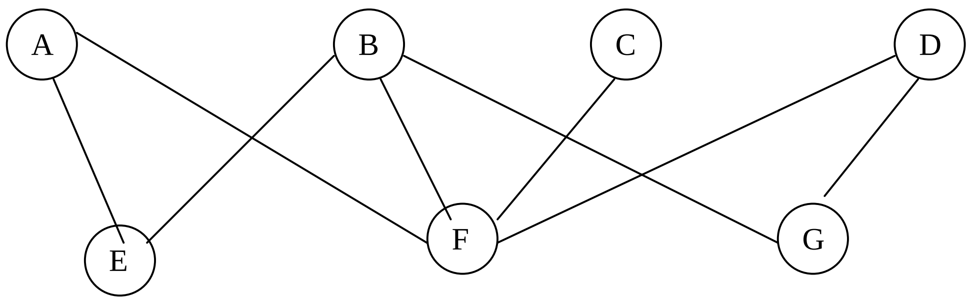
Tree arcs (arcs those make the tree**)**: (0,1), (1,6), (0,2), (2,4), (4,5), (0,3)
Back arcs(an arc from descendent to ancestor) : (2,0), (5,2), (5,4), (5,0)
Cross arcs(an arc neither from ancestor to descendent nor descendent to ancestor)**:** (3,1), (3,5), (3,6), (5,6)
Forward arcs (an arc from ancestor to descendent): (0,6), (0,4)

NB: For an arc (p,q), node p will become ancestor of node q if node p is seen before node q and p is finished after q.
Node p will becomedescendent of node q if node p is seen after node q and p is finished before q.

Example-3: Write the time stamps for all the nodes of the above DFS Tree.

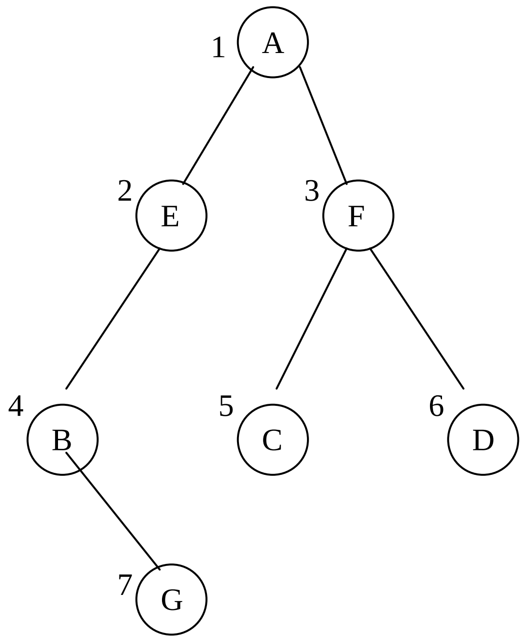
Node	Time Seen	Time Finished
0	0	13
1	1	4
2	5	10
3	11	12
4	6	9
5	7	8
6	2	3

Example-4: What is the order, size, diameter and girth of the following graph.

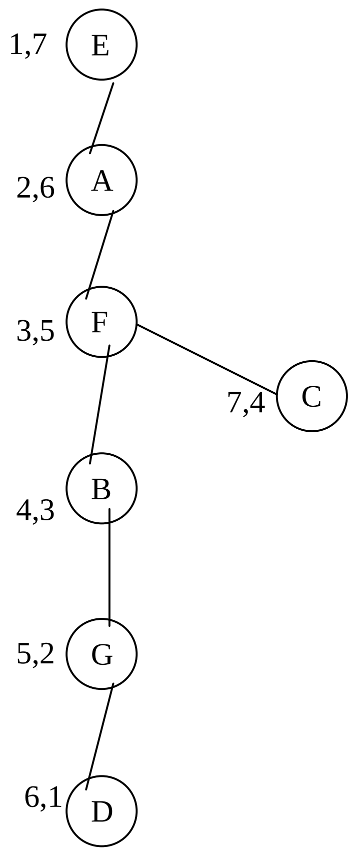


Order (no. of vertex) =7
Size (no. of edges) = 8
Diameter (largest distance between any pair of nodes) = 3
Girth (length of shortest cycle) = 4

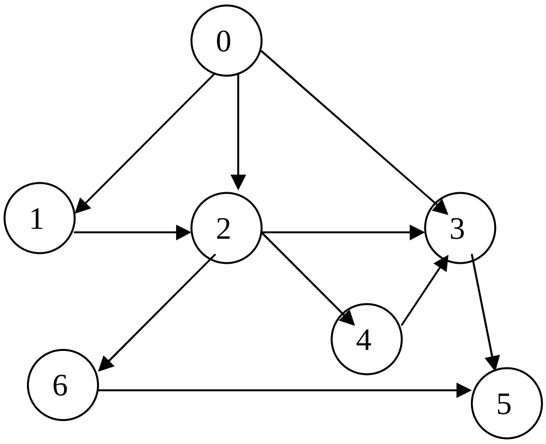
BFS Tree



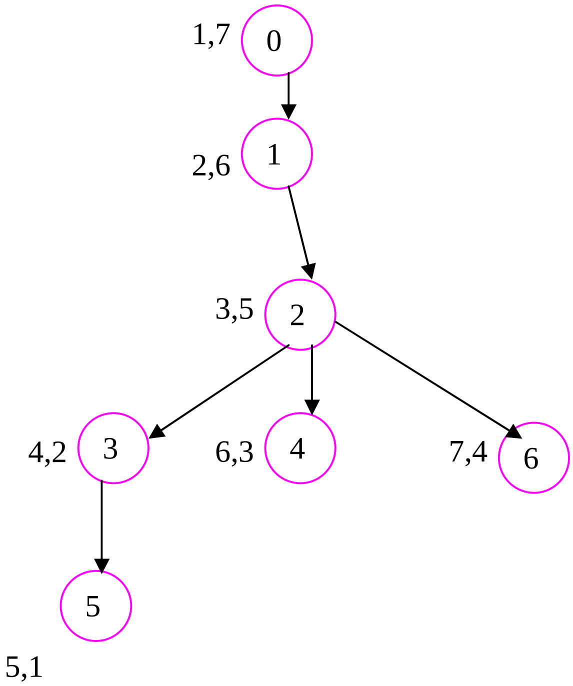
DFS Tree



Example-5: Give the depth-first search (DFS) tree of the following digraph originating from vertex 0 and label the vertices with pre-order and post-order labels.



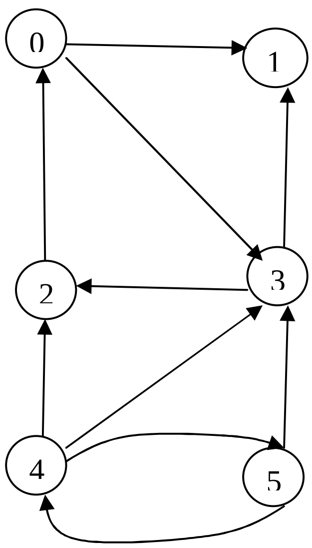
DFS Tree



Example-6: Write down the strongly connected components of the digraph whose adjacency list is given below.

0: 1, 3
1:
2: 0
3: 1, 2
4: 3, 5, 2
5: 3, 4

To find out strongly connected components we need to draw the corresponding digraph of the above adjacency list.



Strongly connected components are: {0, 3, 2}, {4, 5}, {1}

N.B: For finding strongly connected component, first we should look for largest length cycle (i.e to include as many node as possible in the cycle). In this case the cycle {0, 3, 2} Then we should look for next larger length cycle which is in this case {4, 5}. The next one is {1} which includes only one node. We must cover all the nodes. If we can't find any cycle, we have to include single node. But we can't repeat any node.