

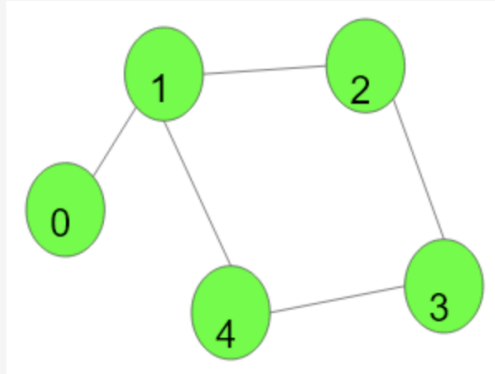
Agenda:

- Detect cycle in an undirected graph
- Detect cycle in a directed graph
- Topological sort
- BFS (shortest path in unweighted graph)
- Dijkstra (shortest path in weighted graph with non-negative weights)

Detect Cycle in Unweighted Graphs:

Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not.

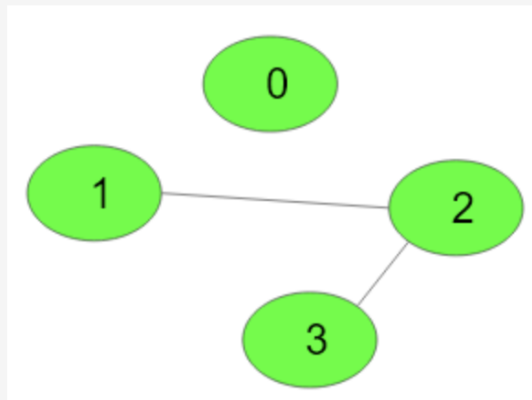
Input:



Output: 1

Explanation: 1->2->3->4->1 is a cycle.

Input:



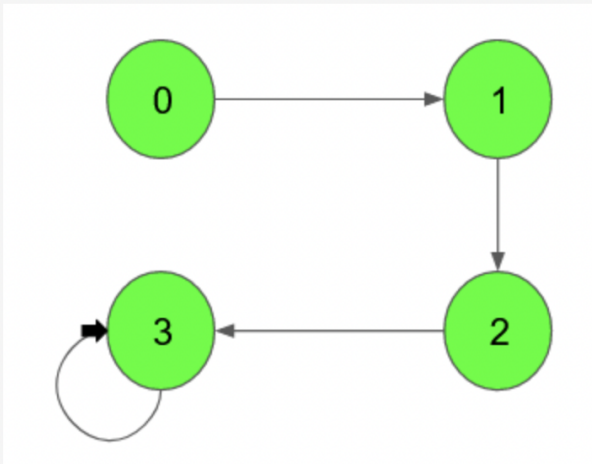
Output: 0

Explanation: No cycle in the graph.

Detect Cycle in a Directed Graph:

Given a Directed Graph with **V** vertices (Numbered from **0** to **V-1**) and **E** edges, check whether it contains any cycle or not.

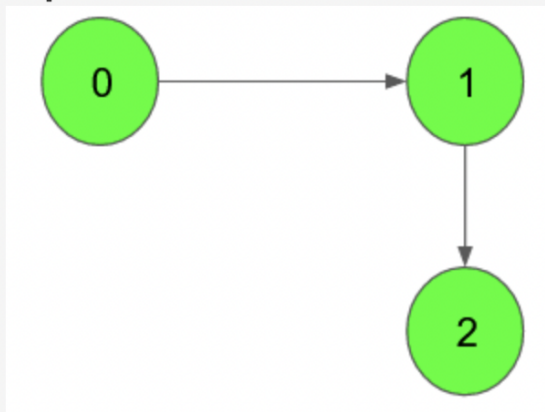
Input:



Output: 1

Explanation: 3 → 3 is a cycle

Input:



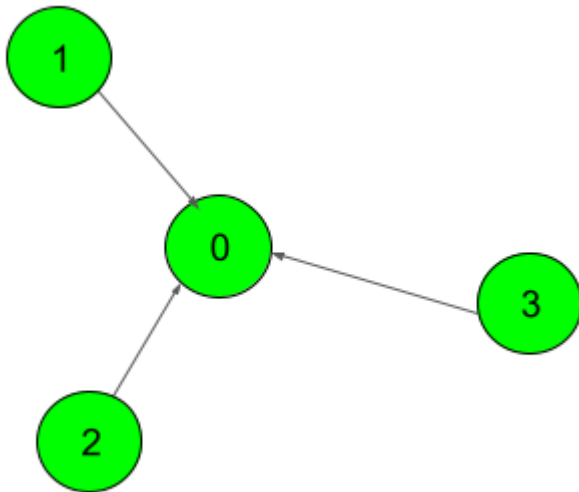
Output: 0

Explanation: no cycle in the graph

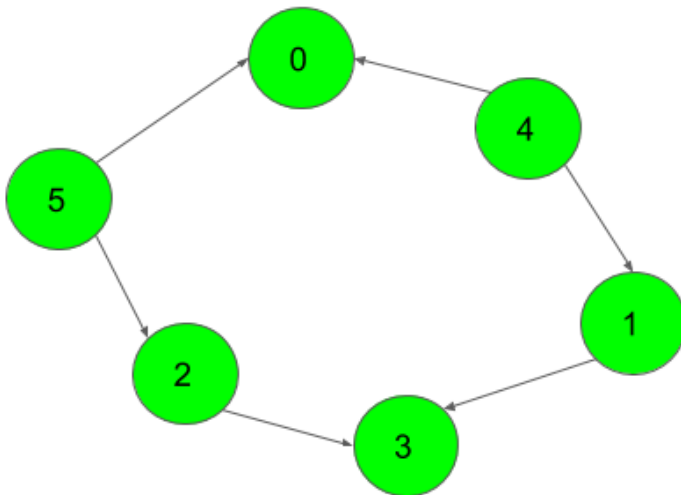
Topological sort:

Given a Directed Acyclic Graph (DAG) with V vertices and E edges, Find any Topological Sorting of that Graph.

Topological sorting for Directed Acyclic Graph (DAG) is a linear ordering of vertices such that for every directed edge $u \rightarrow v$, vertex u comes before v in the ordering.



A correct output can be: [3, 2, 1, 0] or [3, 1, 2, 0] etc.



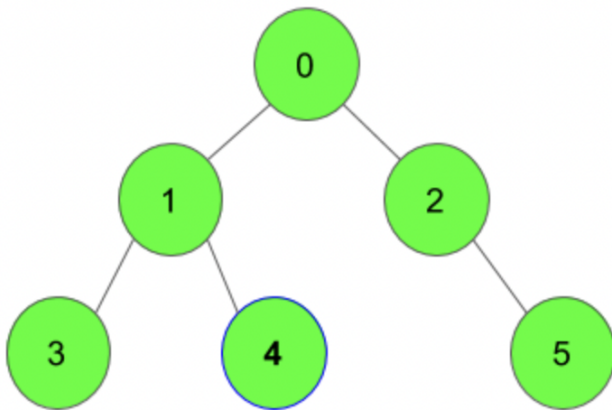
A correct output can be: [5, 4, 2, 1, 3, 0]

BFS (shortest path in unweighted graph)

Given a Undirected Graph with V vertices and E edges, Find the level of node X. if X does not exist in the graph then print -1.

Note: Traverse the graph starting from vertex 0.

Input :

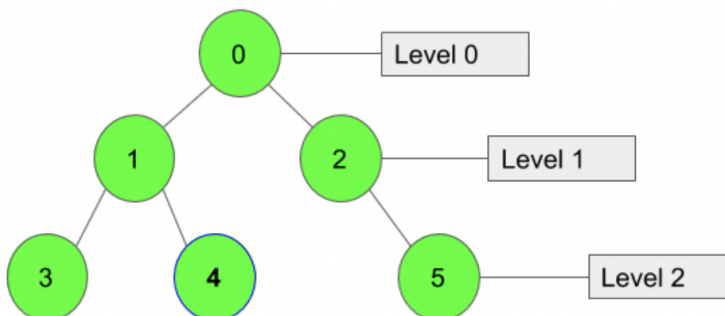


X = 4

Output:

2

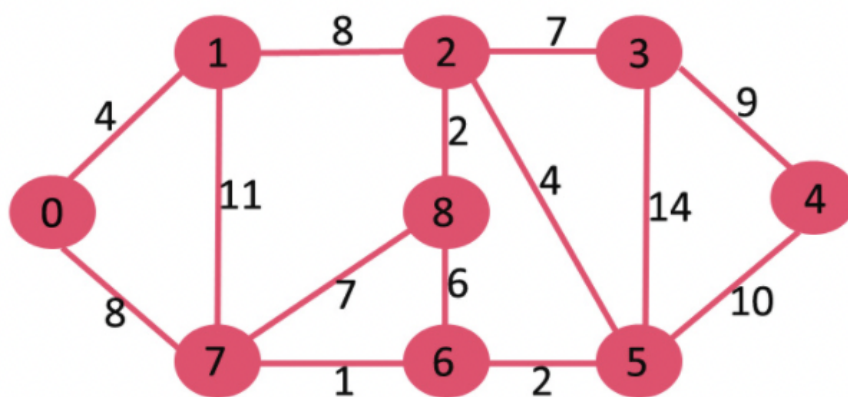
Explanation:



Dijkstra Algorithm:

Dijkstra is an SSSP (Single Source Shortest Path) algorithm used to find the shortest distance from a source node to every other node in a weighted graph.

Note: It works only for a graph that does not have negatively weighted edges.



If source = 0, then:

Vertex	Distance from Source
0	0
1	4
2	12
3	19
4	21
5	11
6	9
7	8
8	14