

# Artificial Intelligence Lab

**Name :** Sandip Kumar Maity

**Roll :** 001911001032

**Department :** Information Technology (UG-3)

**Semester :** 5

**Assignment 3:** Implement iterative deepening

## Program Description :

1. Implement the iterative deepening algorithm
2. Execute the implementation on the attached graph with depth 5
3. Execute the previously implemented DFS with depth 5 on the same graph presented here.
4. For both points 2 and 3, the graph will be written in a file and not in the program implementation.
5. Write a document based on your observations of output 2 and the difference between the outputs of points 2 and 3.

## Observations of Iterative Deepening Algorithm :

1. The Iterative Deepening algorithm performs Depth First Search multiple times with an increasing depth limit from the source node to find the destination node.
2. The Iterative Deepening algorithm searches all nodes up to depth  $i$  in the  $i$ -th iteration.
3. For a particular search, a node may be visited multiple times through different paths.
4. For  $i$ -th iteration, we again traverse all nodes that we traversed in the  $(i - 1)$ -th iteration.
5. The Iterative Deepening algorithm finds the shortest path from the start node to the goal node. If the goal node is found in the  $d$ -th iteration for the first time, a path with length less than  $d$  to the goal node does not exist as the algorithm would have reached the goal through that path in a prior iteration.

## **Differences Between Iterative Deepening vs. DFS with Max Depth Limit :**

<b>Iterative Deepening Algorithm</b>	<b>DFS with Max Depth Limit</b>
1. DFS is executed multiple times with an increasing depth limit.	1. It performs Depth First Search once with a single depth limit from the start node to find the destination node.
2. Here, a node may be visited multiple times through different paths.	2. Here, we visit any node only once.
3. Here, we visit every node up to the given depth limit.	3. Here, we may not visit all nodes at depths up to the max depth limit.
4. The found path is the shortest path to the destination node.	4. Here, if the path to the goal node is found it may not be the shortest path