AURORA DEEMED TO BE UNIVERSITY
AURORA HIGHER EDUCATION AND RESEARCH ACADEMY

| | |
|---|---|
| **Name of the Student: KURUMELI NAGARAJU** | **Academic Year: 2nd Year** |
| **Student Registration Number: 232P4R2061** | **Year & Term: 1st year & 3rd Term** |
| **Study Level: PG** | **Class & Section: MCA DS-B** |
| **Name of the Course: Research and Publication Ethics** | **Name of the Instructor: Maheshprabhu R** |
| **Name of the Assessment: Assignment 2** | **Date of Submission: 20 /07/2025** |

# Assignment 2

## Title: Depth First Search (DFS) Simulation

Aim:To understand and demonstrate how the Depth First Search algorithm works using an interactive simulation.

### Theory:

Depth First Search (DFS) is a fundamental graph traversal technique used to explore nodes and edges of a graph systematically.

- **Working Principle:** DFS starts at a selected source vertex and explores as far along each branch as possible before backtracking.
- **Data Structure Used:** Stack (can be implemented explicitly using a stack or implicitly using recursion).
- **Traversal Strategy:**

1.Visit the current node.

2.Recursively visit all unvisited neighbors.

3.Backtrack when no unvisited neighbors are found.

## Key Characteristics of DFS:

- DFS can be used to detect cycles in a graph.
- It helps in solving maze problems, topological sorting, and connectivity checking.
- DFS is not guaranteed to find the shortest path in weighted graphs.

**Time Complexity:** O(V + E) where V = number of vertices, E = number of edges.
**Space Complexity:** O(V) for recursion/stack storage.

## Procedure:

**1.Open Simulation:**
Access the DFS experiment from *vlab.co.in* → Computer Science and Engineering → Artificial Intelligence → Depth First Search.

**2.Set Parameters:**

- Choose **Start Vertex** (e.g., 4).
- Adjust **Tree Depth** (e.g., 4).
- Keep **AutoPlay** enabled for automated traversal.

**3.Run DFS:**

- Click **Start DFS**.
- Observe how the algorithm visits nodes and builds the frontier.

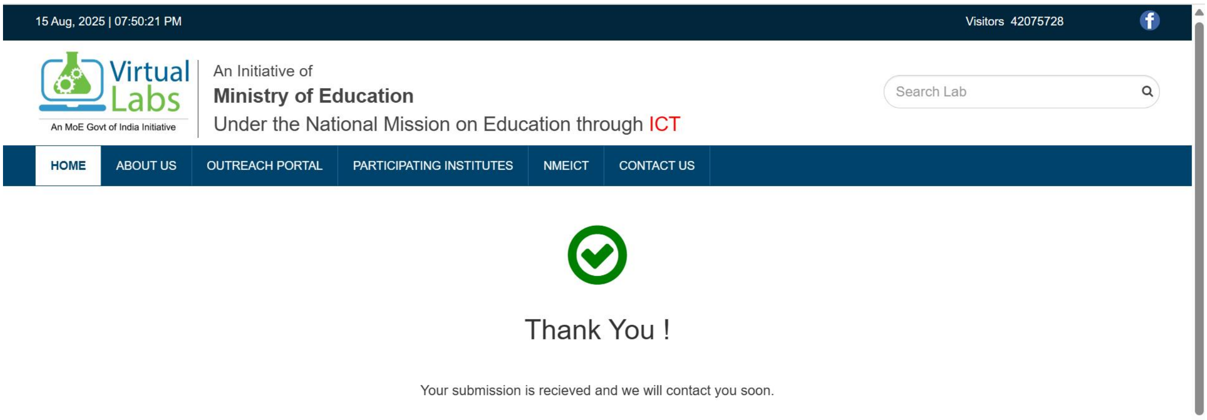
**4.Observation & Recording:**

- Note the visiting order displayed in the *Information* panel.
- Record screenshots for different stages — initial, mid, and final traversal.

## 5.Completion:

- Let the simulation finish visiting all nodes.
- Save final results for assignment submission.

## Simulation Result:



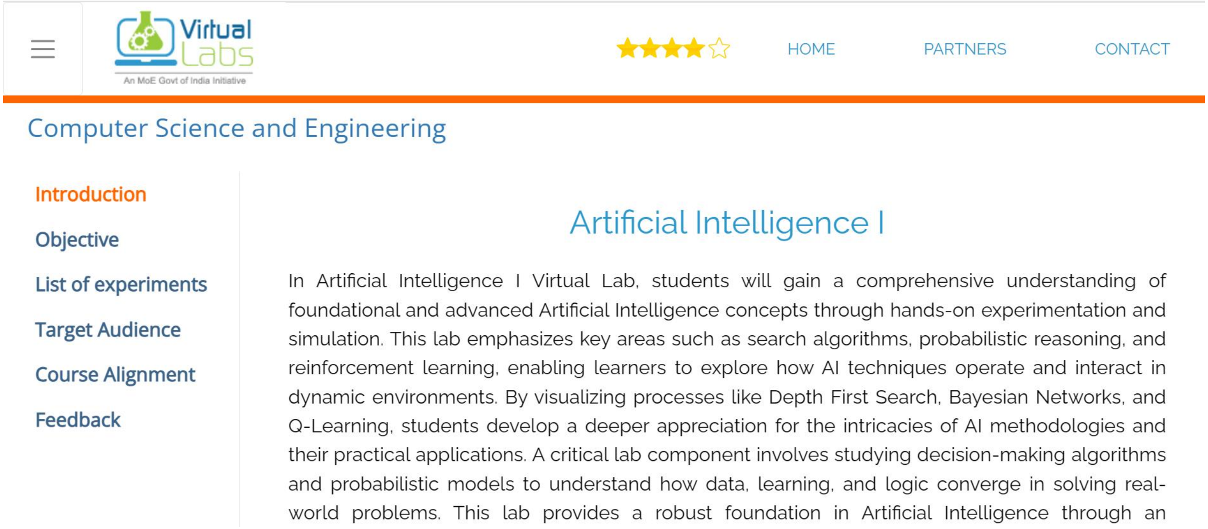
15 Aug, 2025 | 07:50:21 PM

Visitors 42075728

An Initiative of
Ministry of Education
Under the National Mission on Education through ICT

Thank You !

Your submission is recieved and we will contact you soon.



Computer Science and Engineering

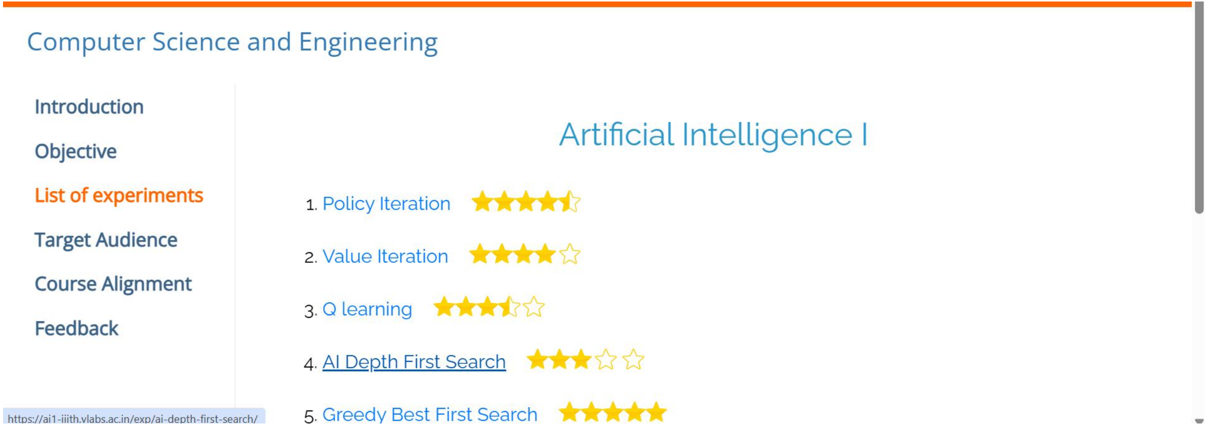
Artificial Intelligence I

In Artificial Intelligence I Virtual Lab, students will gain a comprehensive understanding of foundational and advanced Artificial Intelligence concepts through hands-on experimentation and simulation. This lab emphasizes key areas such as search algorithms, probabilistic reasoning, and reinforcement learning, enabling learners to explore how AI techniques operate and interact in dynamic environments. By visualizing processes like Depth First Search, Bayesian Networks, and Q-Learning, students develop a deeper appreciation for the intricacies of AI methodologies and their practical applications. A critical lab component involves studying decision-making algorithms and probabilistic models to understand how data, learning, and logic converge in solving real-world problems. This lab provides a robust foundation in Artificial Intelligence through an

AURORA DEEMED TO BE UNIVERSITY

AURORA HIGHER EDUCATION AND RESEARCH ACADEMY

aurora TM TEMPLE OF LEARNING



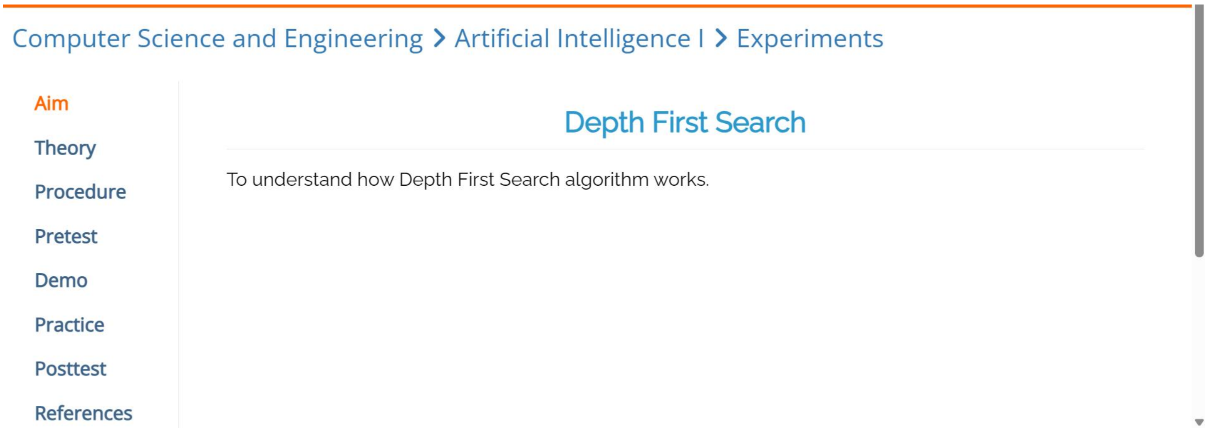
Computer Science and Engineering

- Introduction
- Objective
- List of experiments
- Target Audience
- Course Alignment
- Feedback

https://ai1-iiith.vlabs.ac.in/exp/ai-depth-first-search/

## Artificial Intelligence I

1. Policy Iteration ★★★★⯪
2. Value Iteration ★★★★☆
3. Q learning ★★★⯪☆
4. AI Depth First Search ★★★☆☆
5. Greedy Best First Search ★★★★★

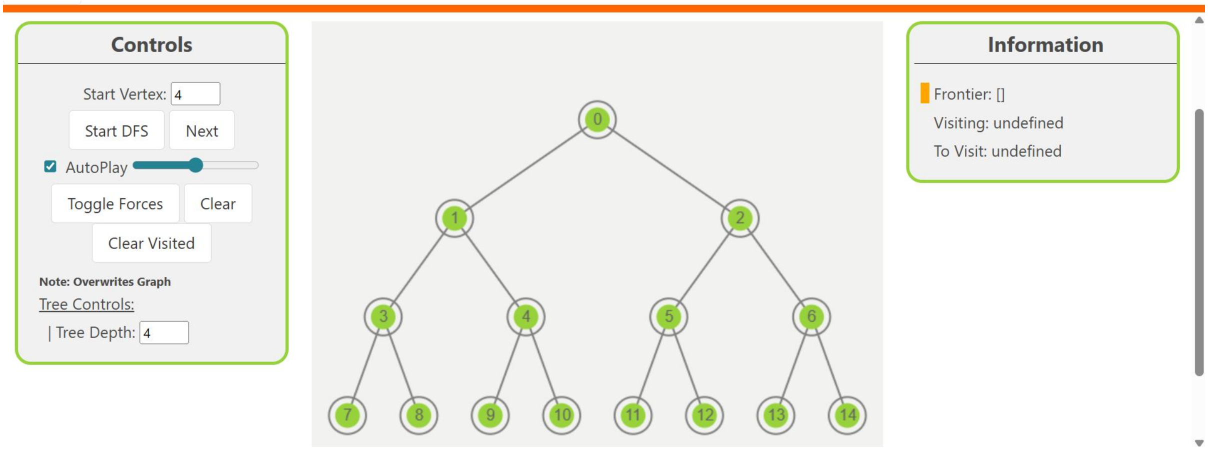


Computer Science and Engineering > Artificial Intelligence I > Experiments

- Aim
- Theory
- Procedure
- Pretest
- Demo
- Practice
- Posttest
- References

## Depth First Search

To understand how Depth First Search algorithm works.



**Controls**

Start Vertex: 4

Start DFS | Next

☑ AutoPlay

Toggle Forces | Clear

Clear Visited

**Note: Overwrites Graph**

Tree Controls:

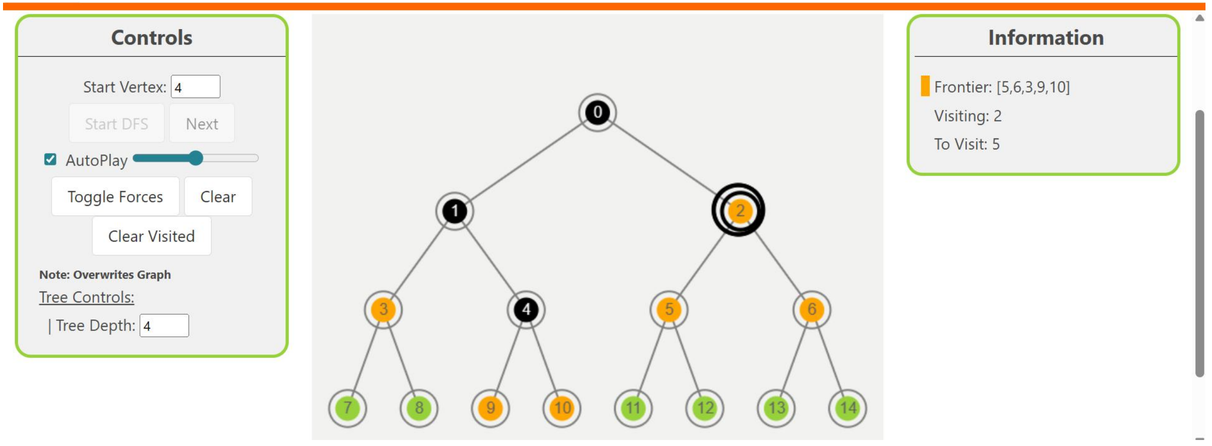
| Tree Depth: 4

**Information**

Frontier: []

Visiting: undefined

To Visit: undefined



Start Vertex: 4

☑ AutoPlay

Note: Overwrites Graph

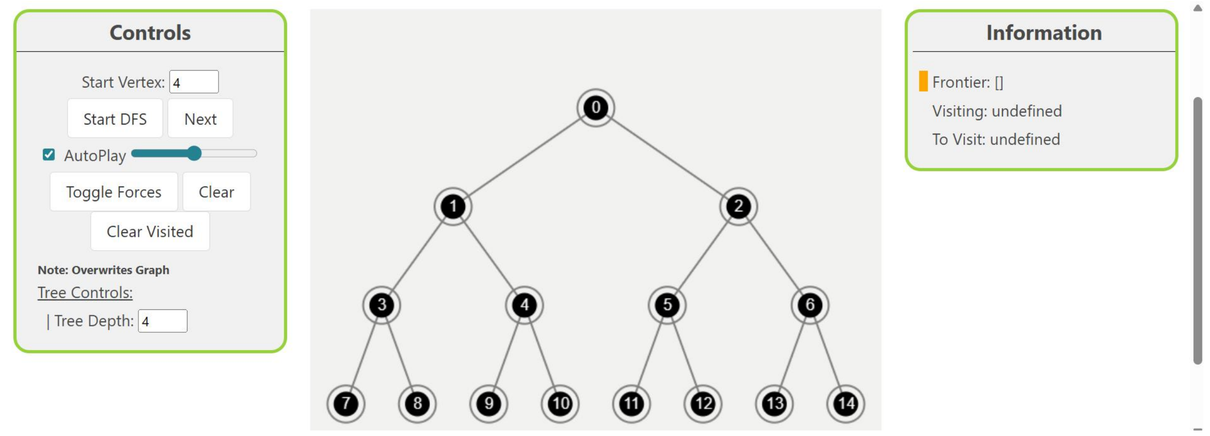
Tree Controls:

| Tree Depth: 4

Frontier: [5,6,3,9,10]

Visiting: 2

To Visit: 5



Start Vertex: 4

☑ AutoPlay

Note: Overwrites Graph

Tree Controls:

| Tree Depth: 4

Frontier: []

Visiting: undefined

To Visit: undefined

## Observation:

From the given tree structure:

- DFS started at vertex **4**.
- The traversal followed depth-wise exploration before moving to sibling branches.
- The *Frontier* list showed nodes queued for future visits, and the *Visiting* field indicated the current node being processed.

## Conclusion:

The DFS algorithm successfully traversed the entire graph in depth-first order. This experiment demonstrated:

- How DFS explores deeply before backtracking.
- The importance of a stack-based approach for traversal order control.
- The difference between DFS and BFS in terms of visiting sequence.