AI PROJECT

Presented By Group 6

Project Overview

In this presentation, we will explore three fundamental search algorithms: Depth-First Search, Iterative Deepening Depth-First Search, and Bidirectional Search. We will delve into their core principles, advantages, and applications, providing a comprehensive understanding of their role in solving complex problems, particularly in the realm of graph theory and artificial intelligence.

Meet The Group

Abhishek Kapoor

Group Leader

Abhinav Goyal

Group Leader

Bhuvan Goyal

Group Leader

Background

Depth-First Search (DFS) explores a graph or tree by traversing as far as possible along a branch before backtracking. Iterative Deepening Depth-First Search (IDDFS) combines the benefits of BFS and DFS by repeatedly increasing the depth of exploration. Bidirectional Search explores from both the start and goal states, meeting in the middle to optimize search in certain situations.

DFS, an early algorithm, dates back to the 19th century. IDDFS was introduced in the 1980s as a space-efficient alternative. Bidirectional Search emerged in the mid-20th century, aiming to reduce search complexity by exploring from both ends.

Contents

Table Of Contents

Chapter 1

What are Search Algorithms?

Search algorithms are fundamental tools in computer science and artificial intelligence. They enable computers to navigate vast datasets or solution spaces to find desired information or solutions. These algorithms form the backbone of various applications, from web search engines to robotics and game-playing AI. At their core, search algorithms determine the most efficient way to explore a problem space and discover the desired outcome. The choice of a specific search algorithm can significantly impact the time and resources required to reach a solution. This introduction will provide insights into the importance, diversity, and practical applications of search algorithms in the digital age.

Iterative Deepening Depth -First Search (IDDFS)

Iterative Deepening Depth-First Search (IDDFS) is a search algorithm that combines the depth-first search strategy with iterative deepening, systematically increasing the depth of exploration in a tree or graph until the desired goal is found. This method ensures completeness, allowing it to explore large solution spaces efficiently while maintaining low memory usage.

Algorithm Steps:

1. Set the initial depth limit to 0.
2. Perform a depth-first search from the start node with the current depth limit.
3. If the goal node is found during this search, return the path to the goal.
4. If the goal node is not found and there are unexplored nodes at the current depth limit, return to step 2.
5. If all nodes at the current depth limit are explored without finding the goal, increment the depth limit by 1.
6. Repeat steps 2-5 until the goal node is found, or the entire search space is exhausted.
7. If the search concludes without finding the goal, the algorithm terminates, indicating that the goal is not present in the graph or tree.

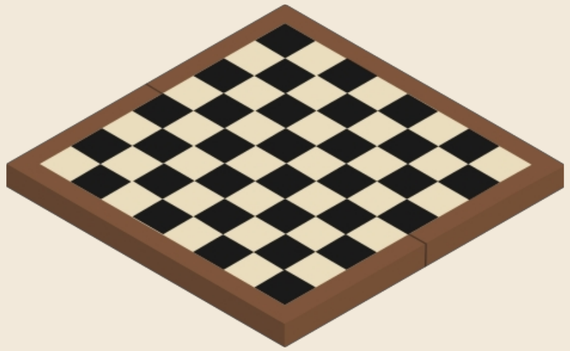
Use Cases:

Puzzle Solving

IDDFS is commonly used in solving puzzles like the N-puzzle (e.g., 8-puzzle, 15-puzzle) or the Tower of Hanoi. It helps find the shortest path to a solution without requiring excessive memory.

Game Playing

In game tree search for games like chess, checkers, or tic-tac-toe, IDDFS can be employed to find the optimal move by exploring different game states within limited memory.



Artificial Intelligence

IDDFS can be applied in AI systems, such as pathfinding in robotics and searching through state spaces in planning problems.

Web Crawling

In web crawling and indexing, IDDFS can be used to explore and navigate web pages while controlling the depth of the crawl to efficiently discover content.

Use Cases:

Natural Language Processing

IDDFS is used in parsing and analyzing syntactic structures in natural language processing to traverse syntax trees and grammars.

Graph Algorithms

IDDFS is useful in various graph algorithms, such as topological sorting, cycle detection, and finding strongly connected components in directed graphs.

DNA Sequence Alignment

In bioinformatics, IDDFS can be applied to align DNA sequences by exploring different alignment possibilities within memory constraints.