A **Depth-First Search (DFS)** is an algorithm used to traverse or search through data structures like trees and graphs. It starts at a root node (or an arbitrary node in the case of a graph) and explores as far down a branch as possible before backtracking to explore other branches. DFS uses a stack data structure, either explicitly with a stack or implicitly with recursion.

Key Points:

- **Exploration**: DFS dives deep into a branch of the tree or graph before moving to the next branch.
- **Stack-Based**: The algorithm can be implemented using a stack or recursion (which uses the call stack).
- **Use Cases**: DFS is used for solving puzzles, pathfinding, topological sorting, and finding connected components in graphs.

DFS is especially useful when you want to explore all possible paths in a structure before making decisions or when memory usage is a concern since it generally uses less memory than breadth-first search (BFS).