**Depth First Search (DFS) Using Stack**

**Introduction**

Depth First Search (DFS) is a graph and tree traversal technique. Unlike Breadth First Search (BFS), which explores level by level, DFS goes as deep as possible along one branch before backtracking. One of the ways to implement DFS is by using an **explicit stack**, which provides a manual way to control the order of traversal without recursion.

**Working Principle**

1. Start from a chosen node (called the source).
2. Place this node into a stack.
3. Repeatedly remove the top node from the stack:
   * If it has not been visited, mark it as visited.
   * Push all of its unvisited neighbors into the stack.
4. Continue until the stack is empty.

This method ensures that the algorithm always follows one path deep into the structure before returning to explore alternative paths.

**Example**

Consider a graph where **A** connects to **B** and **C**, **B** connects to **D** and **E**, and **C** connects to **F**.

* Starting from A, DFS with a stack will visit nodes in the order:  
  **A → C → F → B → E → D** (order may vary based on how neighbors are added).

This shows how DFS goes deep into one side (A → C → F) before returning to explore the rest.

**Advantages**

* Does not require recursion, avoiding recursion depth limits.
* Provides greater control over traversal order (depending on push order).
* Efficient for exploring paths in puzzles, mazes, or backtracking problems.

**Disadvantages**

* Does not guarantee the shortest path in unweighted graphs.
* The stack may grow large for very deep or dense graphs.
* Without a visited check, it can get stuck in infinite loops in cyclic graphs.

**Applications**

* Pathfinding in mazes and puzzles.
* Topological sorting.
* Detecting cycles in graphs.
* Solving backtracking problems (like Sudoku, N-Queens, etc.).

**Conclusion**

DFS using a stack is a powerful way to explore graphs and trees deeply without recursion. While it does not always provide the shortest path, its efficiency in depth exploration makes it suitable for many real-world problems in computer science.