

■ Introduction

■ Core Intuition

- Uses STACK (implicit via recursion, or explicit Stack).
- Goes deep into one branch completely before exploring siblings.
- Good for: cycle detection, topological sort, connected components, path finding.
- NOT guaranteed to find shortest path (unlike BFS).

■ Visual — DFS Traversal

■ Java Implementation

[illegible]

Time & Space Complexity

Case	Time	Space	Why
------	------	-------	-----

Time	$O(V + E)$	$O(V)$	Each vertex and edge processed once
Rec. Space	$O(V)$	$O(V)$	Call stack depth = longest DFS path (up to V)

Common Mistakes

Avoid These

- Not marking visited — causes infinite recursion/loop on cyclic graphs.
- Confusing DFS order with inorder/preorder tree traversal — graph DFS is different.
- Stack overflow on very deep graphs — iterative version is safer.

Practice Problems

Case	Time	Space	Why
Easy	GFG	DFS of Graph	Direct implementation
Medium	LC 200	Number of Islands	DFS on grid
Medium	LC 130	Surrounded Regions	DFS from boundary
Hard	LC 329	Longest Path in Matrix	DFS + memoization

Mental Model

Remember Forever

- "DFS is an explorer who always takes the first available unexplored path, keeps going until a dead end, then backtracks. BFS = ripple in water. DFS = explorer in a maze."