

Detect Cycle — Directed Graph

● MEDIUM

■ Introduction

Cycle detection in DIRECTED graphs is trickier. A visited node in a different DFS branch is NOT a cycle. A back-edge (to an ancestor in the CURRENT DFS path) IS a cycle. We need to track the current recursion stack separately.

■ Core Intuition

Two Arrays Needed

- `visited[]`: marks nodes that have been completely processed (any DFS path).
- `inStack[]`: marks nodes currently in the ACTIVE DFS recursion stack.
- Cycle detected when we reach a neighbor that is in `inStack[]` (ancestor in current path).
- After returning from a node's DFS, remove it from `inStack[]`.

■ Visual

Directed Cycle Visual

Directed Graph: DFS from 0:

0 → 1 → 2 → 3 visit 0, inStack={0}

↑ | visit 1, inStack={0,1}

+-----+ visit 2, inStack={0,1,2}

(3 points back to 1) visit 3, inStack={0,1,2,3}

neighbor of 3 is 1 → 1 is in inStack → CYCLE!

Cross-edge (NOT a cycle):

0→1→3 1 visited from 0's DFS, 3 visited later

0→2→3 (3 reachable from both paths – NOT a cycle)

■ Java Implementation

Java

```
class Solution {
    public boolean isCyclic(int V, ArrayList<ArrayList<Integer>> adj) {
        boolean[] visited = new boolean[V];
        boolean[] inStack = new boolean[V];

        for (int i=0; i<V; i++)
```

```

if (!visited[i] && dfs(i, adj, visited, inStack))
    return true;
return false;
}

private boolean dfs(int node, ArrayList<ArrayList<Integer>> adj,
    boolean[] visited, boolean[] inStack) {
    visited[node] = true;
    inStack[node] = true; // Add to current DFS path

    for (int neighbor : adj.get(node)) {
        if (!visited[neighbor]) {
            if (dfs(neighbor, adj, visited, inStack)) return true;
        } else if (inStack[neighbor]) {
            return true; // Back edge = CYCLE
        }
    }

    inStack[node] = false; // Remove from DFS path on backtrack
    return false;
}
}

```

Time & Space Complexity

Case	Time	Space	Why
Time	$O(V + E)$	$O(V)$	Each vertex and edge once; two $O(V)$ arrays
Space	$O(V)$	$O(V)$	visited + inStack arrays; recursion stack $O(V)$

Common Mistakes

Watch Out

- Confusing with undirected cycle detection — directed needs inStack[], undirected needs parent.
- Forgetting inStack[node]=false on backtrack — causes false cycle detection.
- Using visited[] instead of inStack[] — will falsely flag cross-edges as cycles.

Practice Problems

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

Medium	LC 207	Course Schedule	Detect cycle in directed graph
Medium	LC 210	Course Schedule II	Topological sort (no cycle)
Hard	LC 802	Find Eventual Safe States	Reverse cycle detection