```java
// Function to detect cycle in a directed graph.
public boolean isCyclic(int V, ArrayList<ArrayList<Integer>> adj) {
    boolean[] visited = new boolean[V];
    int[] color = new int[V];
    for (int i = 0; i < V; i++) {
        if (!visited[i]) {
            if (hasCycle(i, adj, visited, color)) {
                return true;
            }
        }
    }
    return false;
}

public boolean hasCycle(int u, ArrayList<ArrayList<Integer>> adj,
        boolean[] visited, int[] color) {
    visited[u] = true;
    color[u] = 1;
    for (int v : adj.get(u)) {
        // if already marked then there contains a cycle
        if (color[v] == 1) {
            return true;
        }
        if (!visited[v] && hasCycle(v, adj, visited, color)) {
            return true;
        }
    }
    // unmark it for next call or handle another forward edge
    color[u] = 0;
    return false;
}
```