**Detect Cycle In A Directed Graph**

bool dfs(int i, vector<vector<int>>& adj, vector<int>& vis, vector<int>& dfsvis) {

vis[i] = 1;

dfsvis[i] = 1;

for (auto child : adj[i]) {

if (!vis[child]) {

if (dfs(child, adj, vis, dfsvis))

return true;

} else if (dfsvis[child]) {

return true;

}

}

dfsvis[i] = 0;

return false;

}

bool detectCycleInDirectedGraph(int n, vector<pair<int, int>>& edges) {

vector<vector<int>> adj(n + 1);

for (int i = 0; i < edges.size(); i++) {

adj[edges[i].first].push\_back(edges[i].second);

}

vector<int> vis(n + 1, 0);

vector<int> dfsvis(n + 1, 0);

for (int i = 1; i <= n; i++) {

if (!vis[i]) {

if (dfs(i, adj, vis, dfsvis))

return true;

}

}

return false;

}