

No of provinces in C++																																																																																									
<pre>#include &lt;bits/stdc++.h&gt; using namespace std;  class Solution { private:     // dfs traversal function     void dfs(int node, vector&lt;int&gt; adjLs[], int vis[]) {         // mark the more as visited         vis[node] = 1;         for(auto it: adjLs[node]) {             if(!vis[it]) {                 dfs(it, adjLs, vis);             }         }     } public:     int numProvinces(vector&lt;vector&lt;int&gt;&gt; adj, int V) {         vector&lt;int&gt; adjLs[V];          // to change adjacency matrix to list         for(int i = 0;i&lt;V;i++) {             for(int j = 0;j&lt;V;j++) {                 // self nodes are not considered                 if(adj[i][j] == 1 &amp;&amp; i != j) {                     adjLs[i].push_back(j);                     adjLs[j].push_back(i);                 }             }         }         int vis[V] = {0};         int cnt = 0;         for(int i = 0;i&lt;V;i++) {             // if the node is not visited             if(!vis[i]) {                 // counter to count the number of provinces                 cnt++;                 dfs(i, adjLs, vis);             }         }         return cnt;     } };  int main() {     vector&lt;vector&lt;int&gt;&gt; adj     {         {1, 0, 1},         {0, 1, 0},         {1, 0, 1}     };      Solution ob;     cout &lt;&lt; ob.numProvinces(adj,3) &lt;&lt; endl;      return 0; }</pre>			<p><b>Input:</b></p> <pre>adj = {     {1, 0, 1},     {0, 1, 0},     {1, 0, 1} }; V = 3</pre> <p>✔ <b>Adjacency Matrix → List Conversion:</b></p> <table> <tr> <th>i</th><th>j</th><th>adj[i][j]</th><th>i != j</th><th>Action</th><th>adjLs</th></tr> <tr> <td>0</td><td>0</td><td>1</td><td>✗</td><td>skip</td><td></td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>✔</td><td>skip</td><td></td></tr> <tr> <td>0</td><td>2</td><td>1</td><td>✔</td><td>add edge 0–2 and 2–0</td><td>0→[2], 2→[0]</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>✔</td><td>skip</td><td></td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>✗</td><td>skip</td><td></td></tr> <tr> <td>1</td><td>2</td><td>0</td><td>✔</td><td>skip</td><td></td></tr> <tr> <td>2</td><td>0</td><td>1</td><td>✔</td><td>already added</td><td></td></tr> <tr> <td>2</td><td>1</td><td>0</td><td>✔</td><td>skip</td><td></td></tr> <tr> <td>2</td><td>2</td><td>1</td><td>✗</td><td>skip</td><td></td></tr> </table> <p>🔗 <b>Final Adjacency List:</b></p> <pre>0 → [2] 1 → [] 2 → [0]</pre> <p>🔗 <b>DFS + Province Counting</b></p> <table> <tr> <th>i</th><th>vis[i]</th><th>Action</th><th>DFS Called</th><th>Updated vis</th><th>cnt</th></tr> <tr> <td>0</td><td>0</td><td>Not visited → DFS(0)</td><td>✔</td><td>[1, 0, 1]</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>Not visited → DFS(1)</td><td>✔</td><td>[1, 1, 1]</td><td>2</td></tr> <tr> <td>2</td><td>1</td><td>Already visited</td><td>✗</td><td>-</td><td>-</td></tr> </table> <p>🔗 <b>DFS Traversal Details</b></p>			i	j	adj[i][j]	i != j	Action	adjLs	0	0	1	✗	skip		0	1	0	✔	skip		0	2	1	✔	add edge 0–2 and 2–0	0→[2], 2→[0]	1	0	0	✔	skip		1	1	1	✗	skip		1	2	0	✔	skip		2	0	1	✔	already added		2	1	0	✔	skip		2	2	1	✗	skip		i	vis[i]	Action	DFS Called	Updated vis	cnt	0	0	Not visited → DFS(0)	✔	[1, 0, 1]	1	1	0	Not visited → DFS(1)	✔	[1, 1, 1]	2	2	1	Already visited	✗	-	-
i	j	adj[i][j]	i != j	Action	adjLs																																																																																				
0	0	1	✗	skip																																																																																					
0	1	0	✔	skip																																																																																					
0	2	1	✔	add edge 0–2 and 2–0	0→[2], 2→[0]																																																																																				
1	0	0	✔	skip																																																																																					
1	1	1	✗	skip																																																																																					
1	2	0	✔	skip																																																																																					
2	0	1	✔	already added																																																																																					
2	1	0	✔	skip																																																																																					
2	2	1	✗	skip																																																																																					
i	vis[i]	Action	DFS Called	Updated vis	cnt																																																																																				
0	0	Not visited → DFS(0)	✔	[1, 0, 1]	1																																																																																				
1	0	Not visited → DFS(1)	✔	[1, 1, 1]	2																																																																																				
2	1	Already visited	✗	-	-																																																																																				

◆ DFS(0)

node	vis[node]	Neighbors	Action	vis
0	0 → 1	2	DFS(2)	[1, 0, 0]
2	0 → 1	0	Already vis	[1, 0, 1]

◆ DFS(1)

node	vis[node]	Neighbors	Action	vis
1	0 → 1	none	Done	[1, 1, 1]

📄 Final Result

Variable	Value
cnt	2 (Answer)
vis	[1, 1, 1]

■ Output: 2 provinces

Output:-  
2