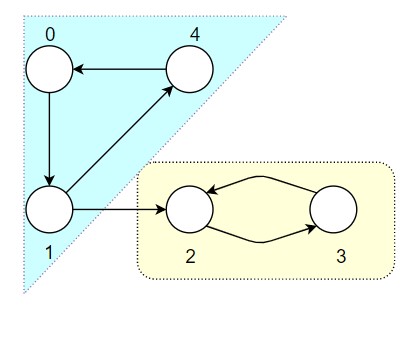
Count Strongly Connected Component using Kosaraju’s Algorithm [CodeStudio](https://www.codingninjas.com/studio/problems/count-strongly-connected-components-kosaraju-s-algorithm_1171151?leftPanelTabValue=PROBLEM)

You are given an unweighted directed graph having 'V' vertices and 'E' edges. Your task is to count the number of strongly connected components (SCCs) present in the graph.

A directed graph is said to be strongly connected if every vertex is reachable from every other vertex. The strongly connected components of a graph are the subgraphs which are themselves strongly connected.

Example:



Output: 2 ({{1, 0, 4}, {2, 3}})

**Approach 1: Function to find strongly connected components using Kosaraju's Algorithm**

* **Explanation:**
  + The algorithm consists of three steps:
    1. Perform DFS traversal and fill finish times in a stack.
    2. Create the transpose of the graph.
    3. Perform reverse DFS traversal on the transpose graph and find strongly connected components.
* **Time Complexity:**
  + **The algorithm performs DFS traversal, transpose creation, and reverse DFS traversal.**
  + **Overall time complexity: O(V + E), where V is the number of vertices and E is the number of edges.**
* **Space Complexity:**
  + **Additional space for the finish times stack: O(V)**
  + **Additional space for the transpose graph: O(E)**
  + **Overall space complexity: O(V + E)**