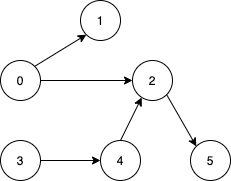
Minimum Number of Vertices to Reach All Nodes [LeetCode](https://leetcode.com/problems/minimum-number-of-vertices-to-reach-all-nodes/description/)

Given a**directed acyclic graph**, with n vertices numbered from 0 to n-1, and an array edges where edges[i] = [fromi, toi] represents a directed edge from node fromi to node toi.

Find *the smallest set of vertices from which all nodes in the graph are reachable*. It's guaranteed that a unique solution exists.

Notice that you can return the vertices in any order.

Example:



Output: {0, 3}

Example 1:

A group of white circles with black text

Description automatically generated

Output: {0, 2, 3}

**Approach 1: Find the minimum vertices to reach all nodes using Indegree count**

* **Explanation:**
  + The **findSmallestSetOfVertices** function calculates the in-degrees of each vertex based on the given edges.
  + Vertices with in-degree 0 are identified as those not reachable by any other vertices.
* **Time Complexity:**
  + **The time complexity is O(V + E), where V is the number of vertices and E is the number of edges in the graph.**
    - **Calculating in-degrees for each edge.**
* **Space Complexity:**
  + **The space complexity is O(V), where V is the number of vertices in the graph.**
    - **Storing in-degree information for each vertex.**