5480. Minimum Number of Vertices to Reach All Nodes

omissions (/contest/biweekly-contest-33/problems/minimum-number-of-vertices-to-reach-all-nodes/submissions/)

Contest (/contest/biweekly-contest-33/)

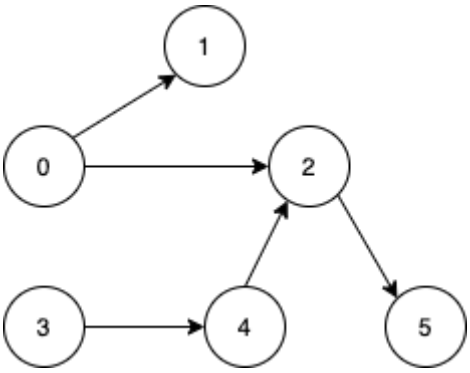
Given a **directed acyclic graph**, with `n` vertices numbered from `0` to `n-1`, and an array `edges` where `edges[i] = [from_i, to_i]` represents a directed edge from node `from_i` to node `to_i`.

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.

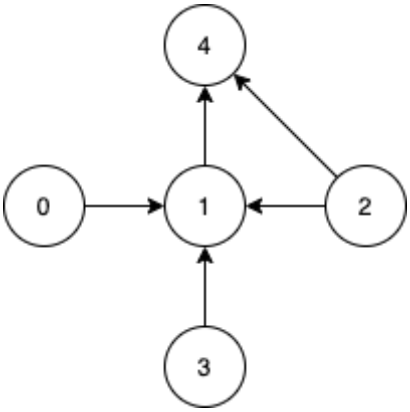
User Accepted:	2463
User Tried:	2851
Total Accepted:	2499
Total Submissions:	3536
Difficulty:	Medium

Example 1:




```
Input: n = 6, edges = [[0,1],[0,2],[2,5],[3,4],[4,2]]
Output: [0,3]
Explanation: It's not possible to reach all the nodes from a single vertex. From 0 we can
```

Example 2:



```
Input: n = 5, edges = [[0,1],[2,1],[3,1],[1,4],[2,4]]
Output: [0,2,3]
Explanation: Notice that vertices 0, 3 and 2 are not reachable from any other node, so we
```

Constraints:

- 2 <= n <= 10^5
- 1 <= edges.length <= min(10^5, n * (n - 1) / 2)
- edges[i].length == 2
- 0 <= $from_i$, to_i < n
- All pairs ($from_i$, to_i) are distinct.

JavaScript

```
/**
 * @param {number} n
 * @param {number[][]} edges
 * @return {number[]}
 */
var findSmallestSetOfVertices = function(n, edges) {

};
```

Custom Testcase Use Example Testcases

Run Submit