**1483. Kth Ancestor of a Tree Node**

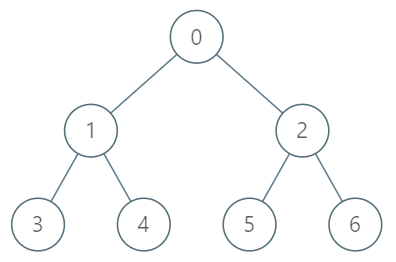
Hard

You are given a tree with n nodes numbered from 0 to n-1 in the form of a parent array where parent[i] is the parent of node i. The root of the tree is node 0.

Implement the function getKthAncestor(int node, int k) to return the k-th ancestor of the given node. If there is no such ancestor, return -1.

The *k-th ancestor* of a tree node is the k-th node in the path from that node to the root.

**Example:**

****

**Input:**

["TreeAncestor","getKthAncestor","getKthAncestor","getKthAncestor"]

[[7,[-1,0,0,1,1,2,2]],[3,1],[5,2],[6,3]]

**Output:**

[null,1,0,-1]

**Explanation:**

TreeAncestor treeAncestor = new TreeAncestor(7, [-1, 0, 0, 1, 1, 2, 2]);

treeAncestor.getKthAncestor(3, 1); // returns 1 which is the parent of 3

treeAncestor.getKthAncestor(5, 2); // returns 0 which is the grandparent of 5

treeAncestor.getKthAncestor(6, 3); // returns -1 because there is no such ancestor

**Constraints:**

* 1 <= k <= n <= 5\*10^4
* parent[0] == -1 indicating that 0 is the root node.
* 0 <= parent[i] < n for all 0 < i < n
* 0 <= node < n
* There will be at most 5\*10^4 queries.

Accepted

9,418

Submissions

31,467