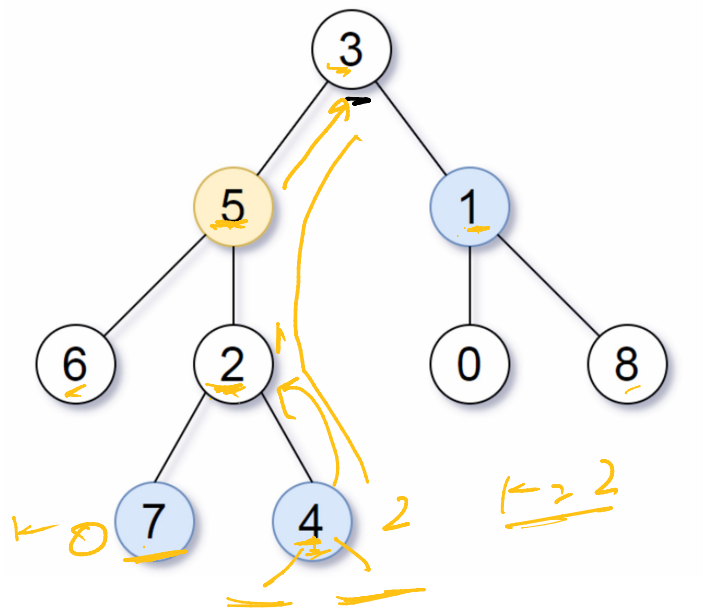
28-Jan

28 January 2022 19:08

All nodes at K-dist

https://leetcode.com/problems/all-nodes-distance-k-in-binary-tree/



tar = 5

k = 2

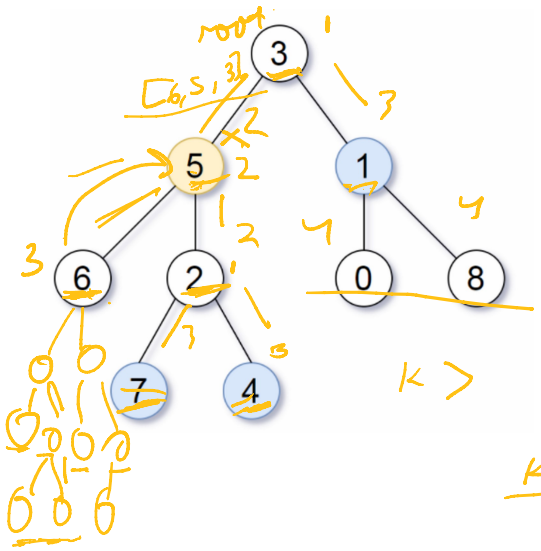
[5, 3]

[4, 2, 5, 3] k = 3

k=0 k=1 k=2 k=3

(k - i < 0)

tar = 6 , k = 3




```
public static ArrayList<Node> nodeToRootPath(Node root, Node tar) {
    if (root == tar) {
        return new ArrayList<Node>(Collections.singletonList(root));
    }
    ArrayList<Node> left = root.left == null ? null : nodeToRootPath(root.left, tar);
    if (left != null) {
        left.add(root);
        return left;
    }
    ArrayList<Node> right = root.right == null ? null : nodeToRootPath(root.right, tar);
    if (right != null) right.add(root);
    return right;
}
```

Step 1 - find node to root path

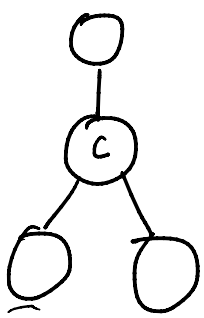
[6, 5, 3]

k = 2

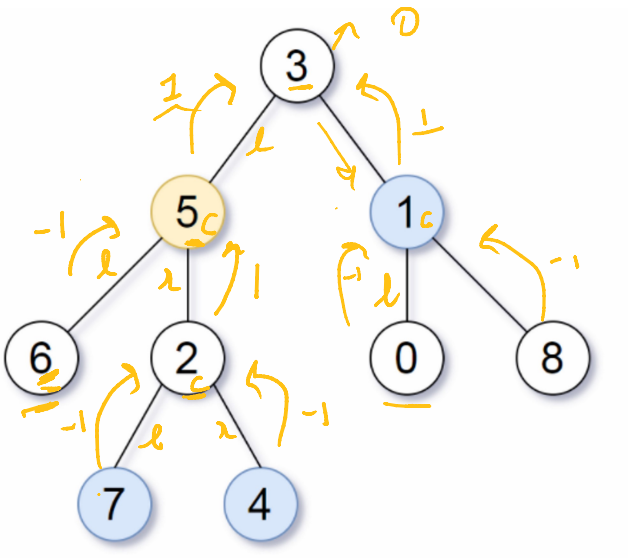
k = 1

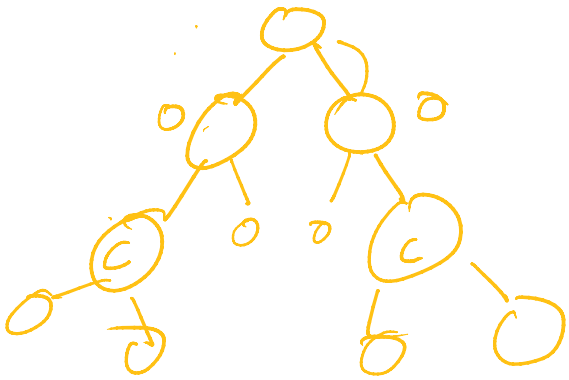
k = 0

Cameras in Binary trees -

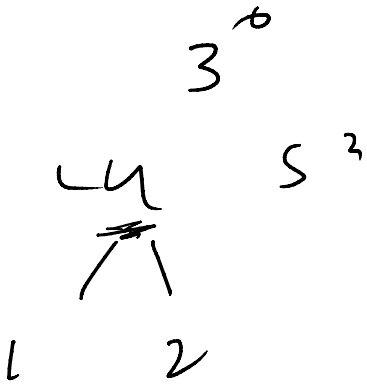


- Camera has placed = 1
- Node is under the camera = 0
- No camera can be seen around = -1





Subtree in a tree :- https://leetcode.com/problems/subtree-of-another-tree/



```
boolean sot (root, sub) {
  if(root == null) return false
  if(root == sub) return check(root, sub);
  left = sot(root.left, sub)
  right = sot(root.right, sub);
  return (left || right);
}

boolean check(root, sub) {
  if(root != sub) return false
  left = check(root.left, sub.left)
  right = check(root.right, sub.right);
  return left && right;
}
```