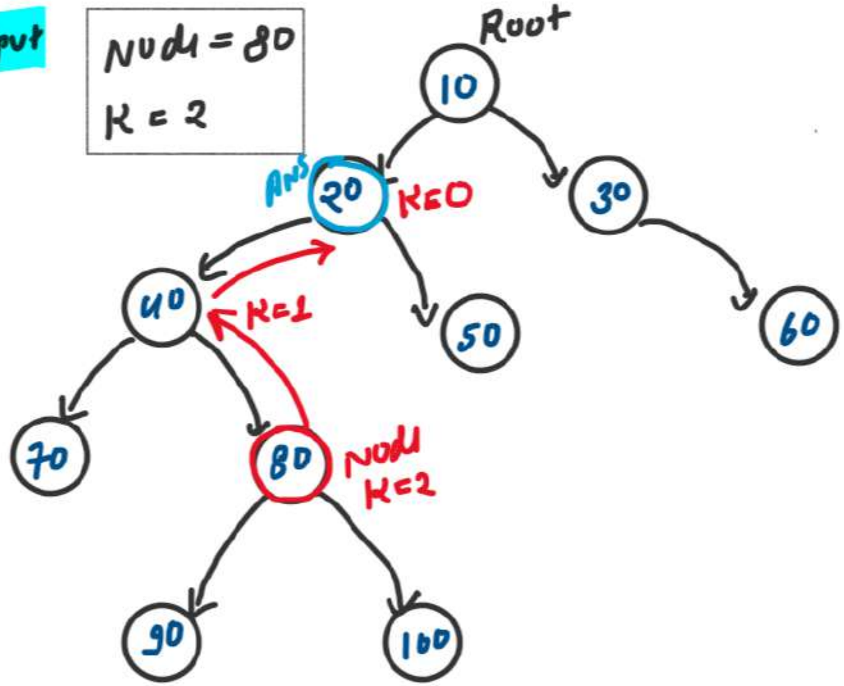
1. Kth Ancestor of a Node in Binary Tree (GFG)

Input

Node = 80
K = 2



Output = 20

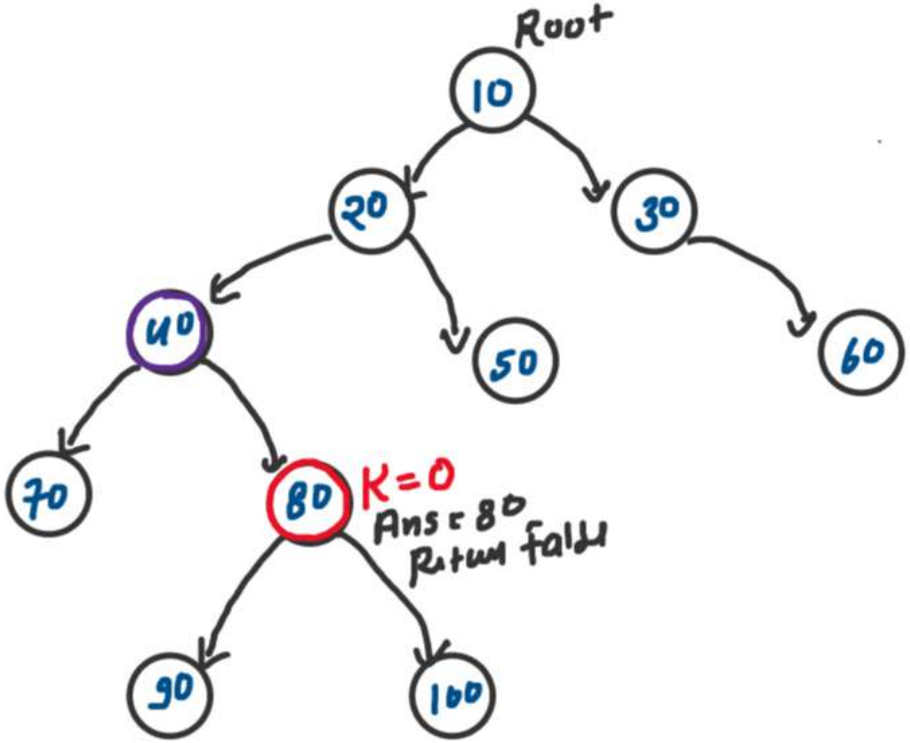
Approach

STEP1 First Find the given Node = 80

STEP2 Traversal back from found Node location Jab tak K==0 Na Ho

STEP3 if(K==0)
↳ Return Root Node→val

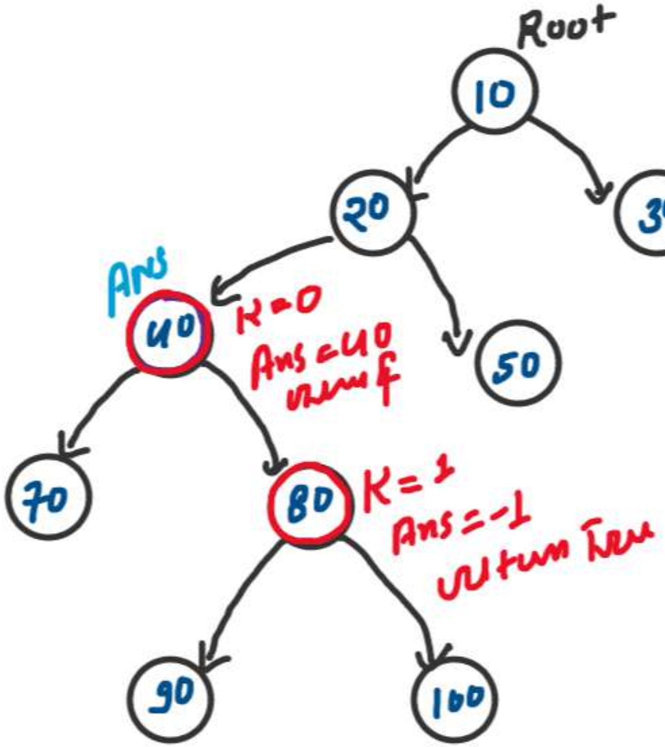
DRY RUN



K = 0   K = 80   Ans = 80

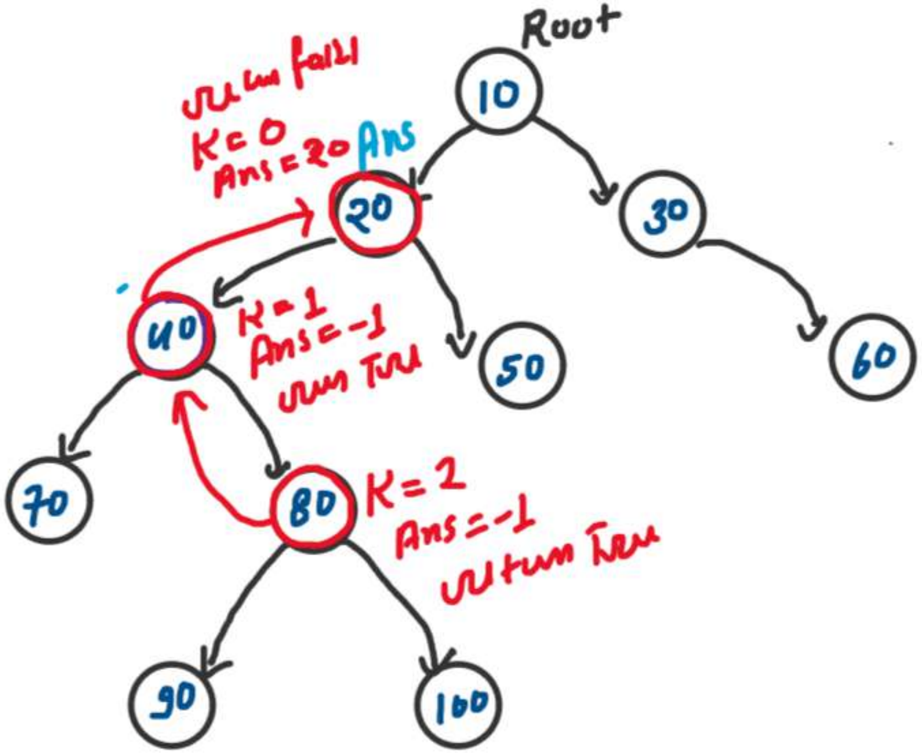
DRY RUN

K = 1 K = 80 Ans = 40

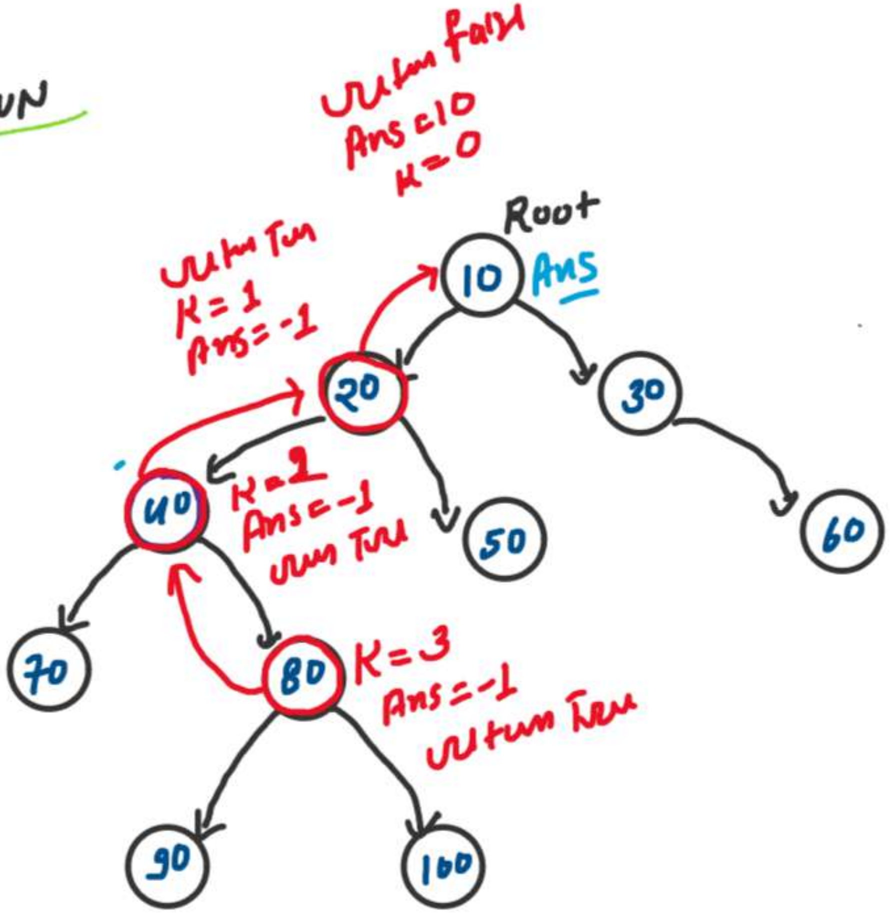


DRY RUN

K = 2 K = 80 Ans = 20

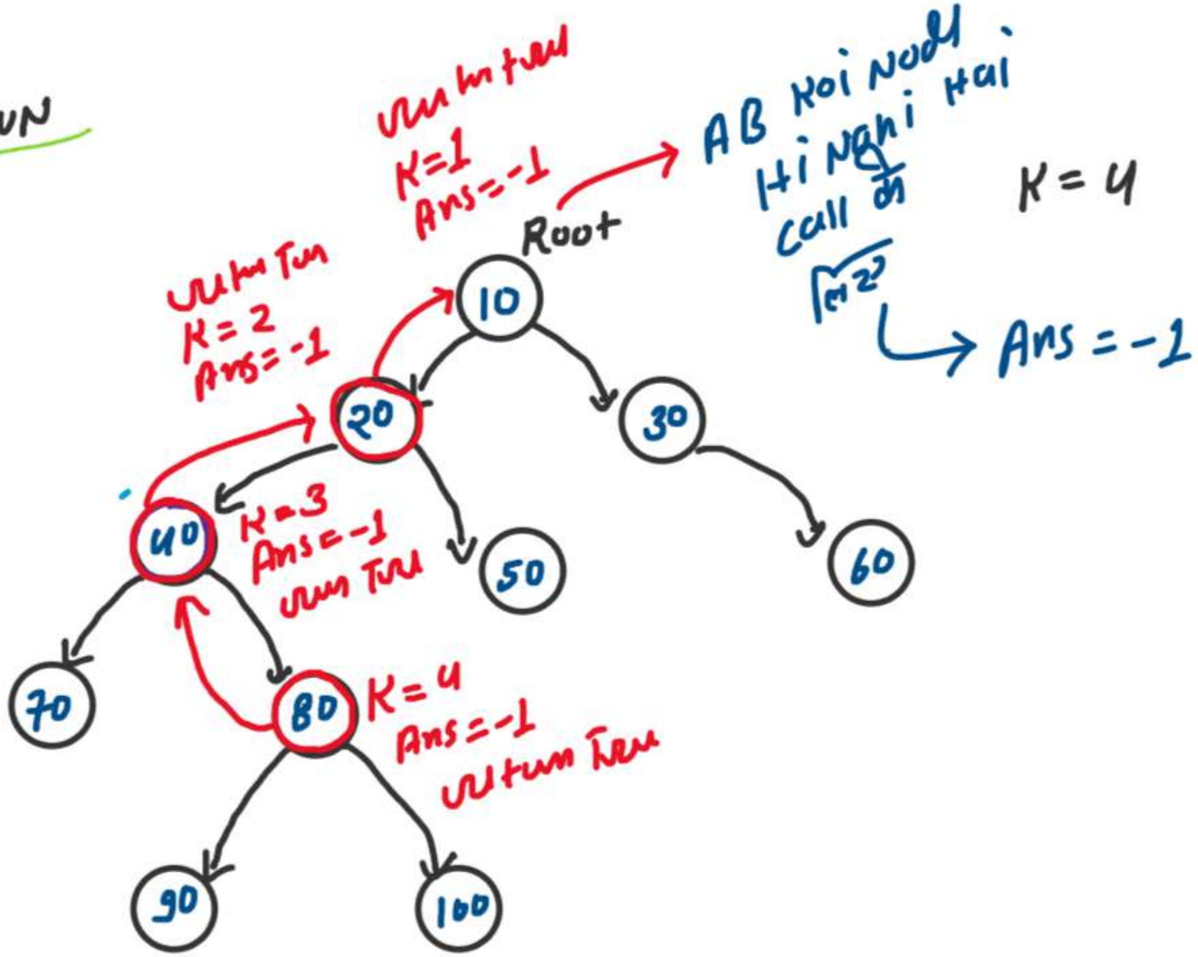


DRY RUN

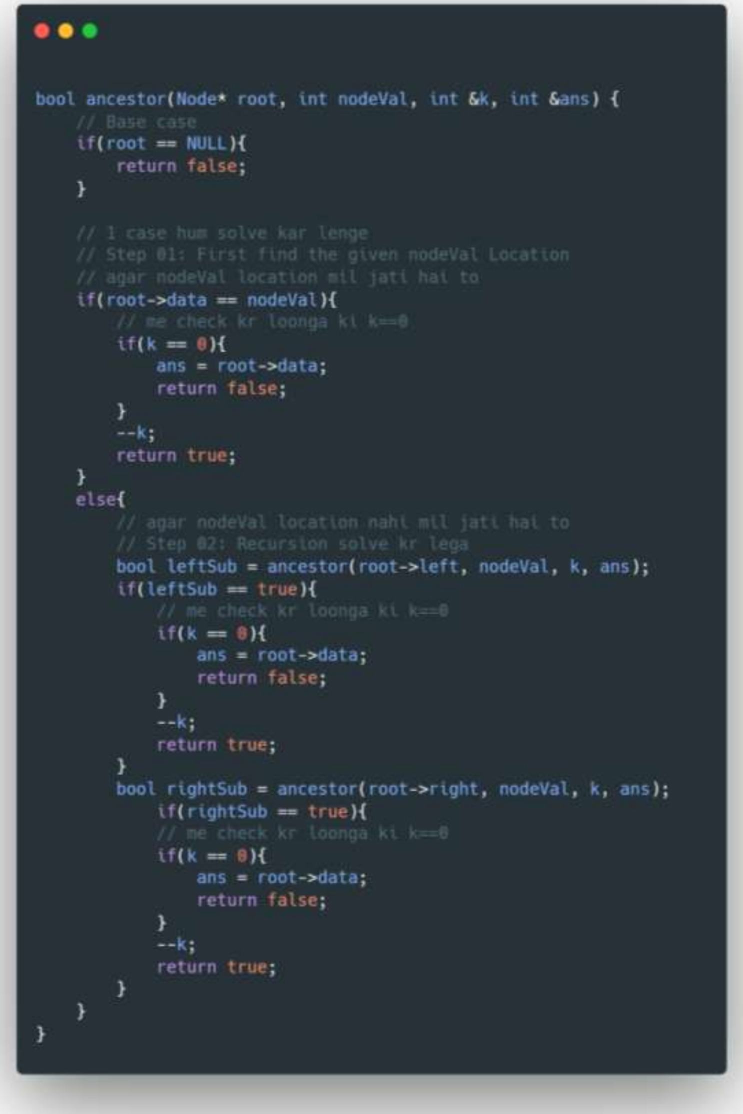


K = 3 K = 80 Ans = 10

DRY RUN



K = 4   K = 80   Ans = -1



```cpp
bool ancestor(Node* root, int nodeVal, int &k, int &ans) {
    // Base case
    if(root == NULL){
        return false;
    }

    // 1 case hum solve kar lenge
    // Step 01: First find the given nodeVal Location
    // agar nodeVal location mil jati hai to
    if(root->data == nodeVal){
        // me check kr loonga ki k==0
        if(k == 0){
            ans = root->data;
            return false;
        }
        --k;
        return true;
    }
    else{
        // agar nodeVal location nahi mil jati hai to
        // Step 02: Recursion solve kr lega
        bool leftSub = ancestor(root->left, nodeVal, k, ans);
        if(leftSub == true){
            // me check kr loonga ki k==0
            if(k == 0){
                ans = root->data;
                return false;
            }
            --k;
            return true;
        }
        bool rightSub = ancestor(root->right, nodeVal, k, ans);
            if(rightSub == true){
            // me check kr loonga ki k==0
            if(k == 0){
                ans = root->data;
                return false;
            }
            --k;
            return true;
        }
    }
}
```

T.C. and S.C. =?