**SOLUTION**

class Solution {

public:

bool isCousins(TreeNode\* root, int x, int y) {

int px=0;

int py=0;

int dx=-1;

int dy=-1;

bfs(root,x,y,0,dx,dy,px,py);

return ((dx==dy) && (px!=py));

}

void bfs(TreeNode\* root, int x, int y, int d, int& dx, int& dy, int& px, int& py){

if(!root)

return;

if(root->left){

if(root->left->val==x){

px=root->val;

dx=d+1;

} else if(root->left->val==y){

py=root->val;

dy=d+1;

}

}

if(root->right){

if(root->right->val==x){

px=root->val;

dx=d+1;

} else if(root->right->val==y){

py=root->val;

dy=d+1;

}

}

if(dx!=-1 && dy!=-1) return;

bfs(root->left,x,y,d+1,dx,dy,px,py);

bfs(root->right,x,y,d+1,dx,dy,px,py);

}

};

**TIME COMPLEXITY: O(N)**

**SPACE COMPLEXITY: O(1)**