

Follow me for more interesting programming questions and RAW Code visit my GitHub profile:
<https://www.github.com/shubhamthrills> <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCjLMu9mayAibQST1eWwq0cg>

Leetcode May Challenge DAY: 20

1. Python

```
stack = []
```

```
while True:
```

```
    while root:
```

```
        stack.append(root)
```

```
        root = root.left
```

```
    root = stack.pop()
```

```
    k -= 1
```

```
    if not k:
```

```
        return root.val
```

```
    root = root.right
```



Follow me for more interesting programming questions and RAW Code visit my GitHub profile:
<https://www.github.com/shubhamthrills> <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCjLMu9mayAibQST1eWwq0cg>

2. C++

```
class Solution {
```

```
public:
```

```
    int ans, count = 0;
```

```
    void inorder(TreeNode* root, int k){
```

```
        if(root == nullptr) return ;
```

```
        inorder(root->left, k);
```

```
        count++;
```

```
        if(count == k){
```

```
            ans = root->val;
```

```
            return ;
```

```
        }
```

```
        inorder(root->right, k);
```

```
    }
```

```
    int kthSmallest(TreeNode* root, int k) {
```

```
        inorder(root, k);
```

```
        return ans;
```

```
    }
```

```
};
```

Follow me for more interesting programming questions and RAW Code visit my GitHub profile:
<https://www.github.com/shubhamthrills> <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCjLMu9mayAibQST1eWwq0cg>

3. JAVA

```
class Solution {  
    int ans, count;  
    public void inorder(TreeNode root, int k){  
        if(root == null) return ;  
        inorder(root.left, k);  
        count++;  
        if(count == k){  
            ans = root.val;  
            return ;  
        }  
        inorder(root.right, k);  
    }  
    public int kthSmallest(TreeNode root, int k) {  
        ans = 0;  
        inorder(root, k);  
        return ans;  
    }  
}
```