Binary Search Tree (2/3): Traversal

Shusen Wang

Stevens Institute of Technology

http://wangshusen.github.io/

Print all the keys

Tree:

```
        50
      /     \
    30       70
   /  \        \
20    40        80
```

Print all the keys

Tree:

```
        50
     /      \
    30        70
   /  \         \
20    40        80
```

Print:

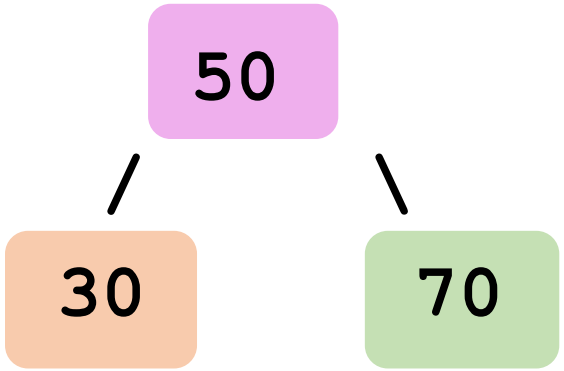
```
20
30
40
50
70
80
```

Print all the keys

Tree:	Print:
50	50

Print all the keys

Tree:

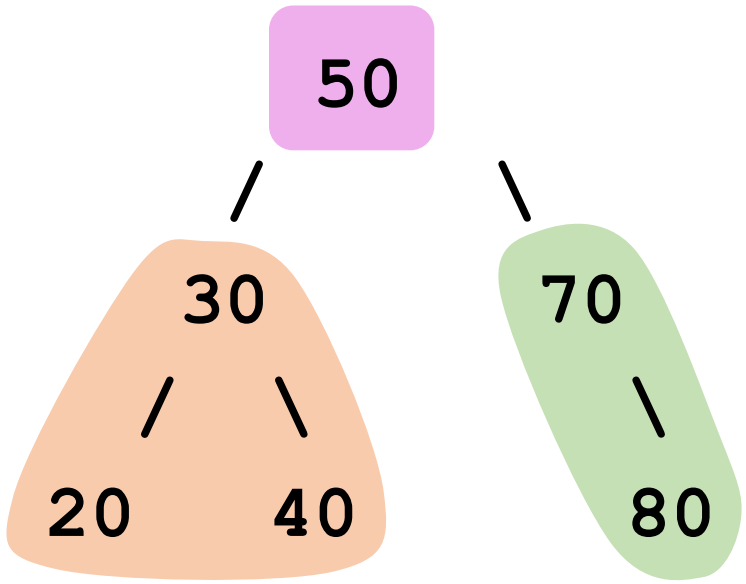


Print:

30
50
70

Print all the keys

Tree:



Print:

20
30
40
50
70
80

Print all the keys

```
void traverse(struct vertex *root) {
    if (root != NULL) {
        traverse(root->left);
        cout << root->key << endl;
        traverse(root->right);
    }
}
```

Print all the keys

```
void traverse(struct vertex *root) {
    if (root != NULL) {
        traverse(root->left);
        cout << root->key << endl;
        traverse(root->right);
    }
}
```

Print all the keys

```
void traverse(struct vertex *root) {
    if (root != NULL) {
        traverse(root->left);
        cout << root->key << endl;
        traverse(root->right);
    }
}
```

Inorder, Preorder, and Postorder

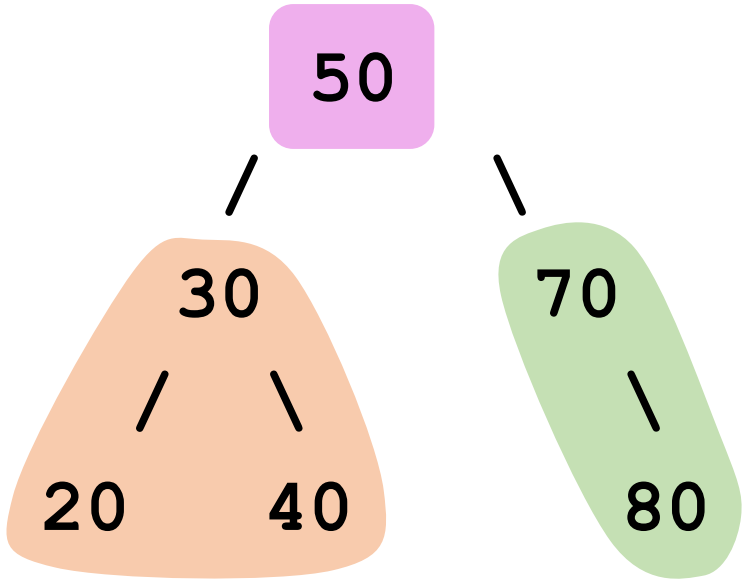
Inorder Traversal

This is what we learned in this class.

```
void inorder(struct vertex *root) {
    if (root != NULL) {
        inorder(root->left);
        cout << root->key << endl;
        inorder(root->right);
    }
}
```

Inorder Traversal

Tree:



Inorder print:

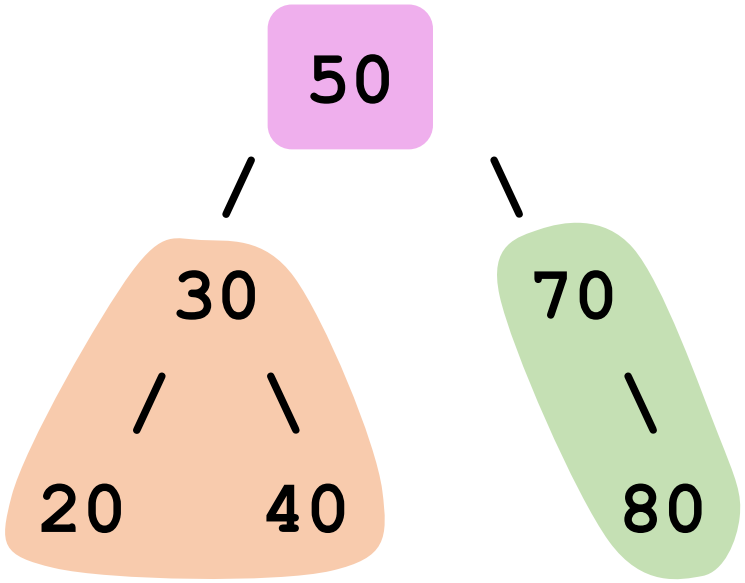
20
30
40
50
70
80

Preorder Traversal

```
void preorder(struct vertex *root) {
    if (root != NULL) {
        cout << root->key << endl;
        preorder(root->left);
        preorder(root->right);
    }
}
```

Preorder Traversal

Tree:



Preorder print:

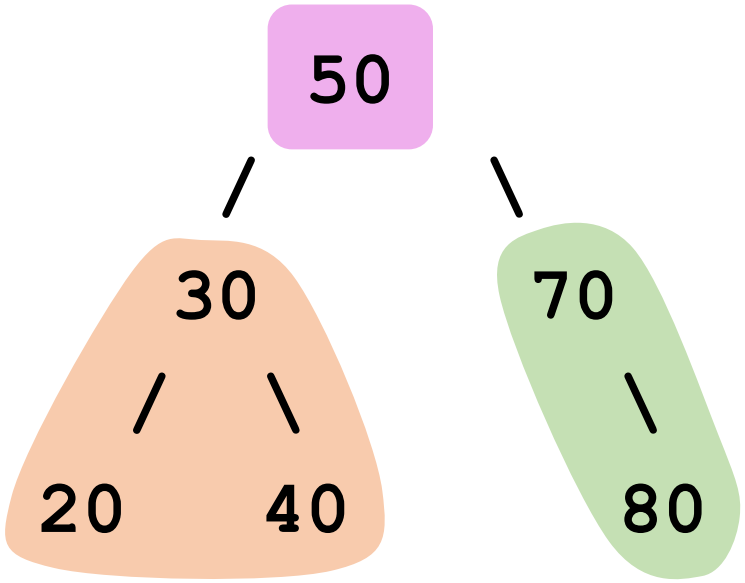
50
30
20
40
70
80

Postorder Traversal

```
void postorder(struct vertex *root) {
    if (root != NULL) {
        postorder(root->left);
        postorder(root->right);
        cout << root->key << endl;
    }
}
```

Postorder Traversal

Tree:

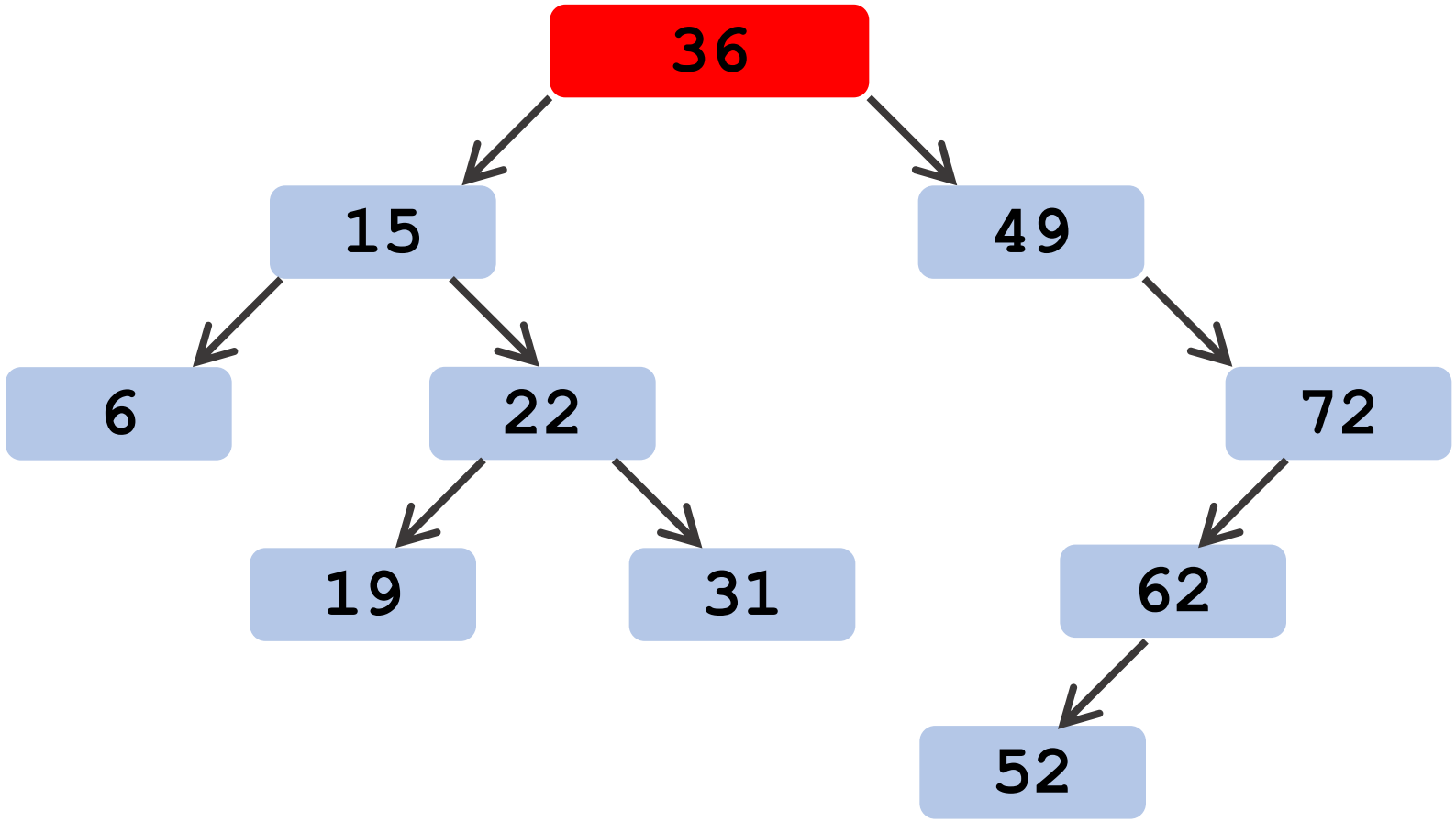


Postorder print:

20
40
30
80
70
50

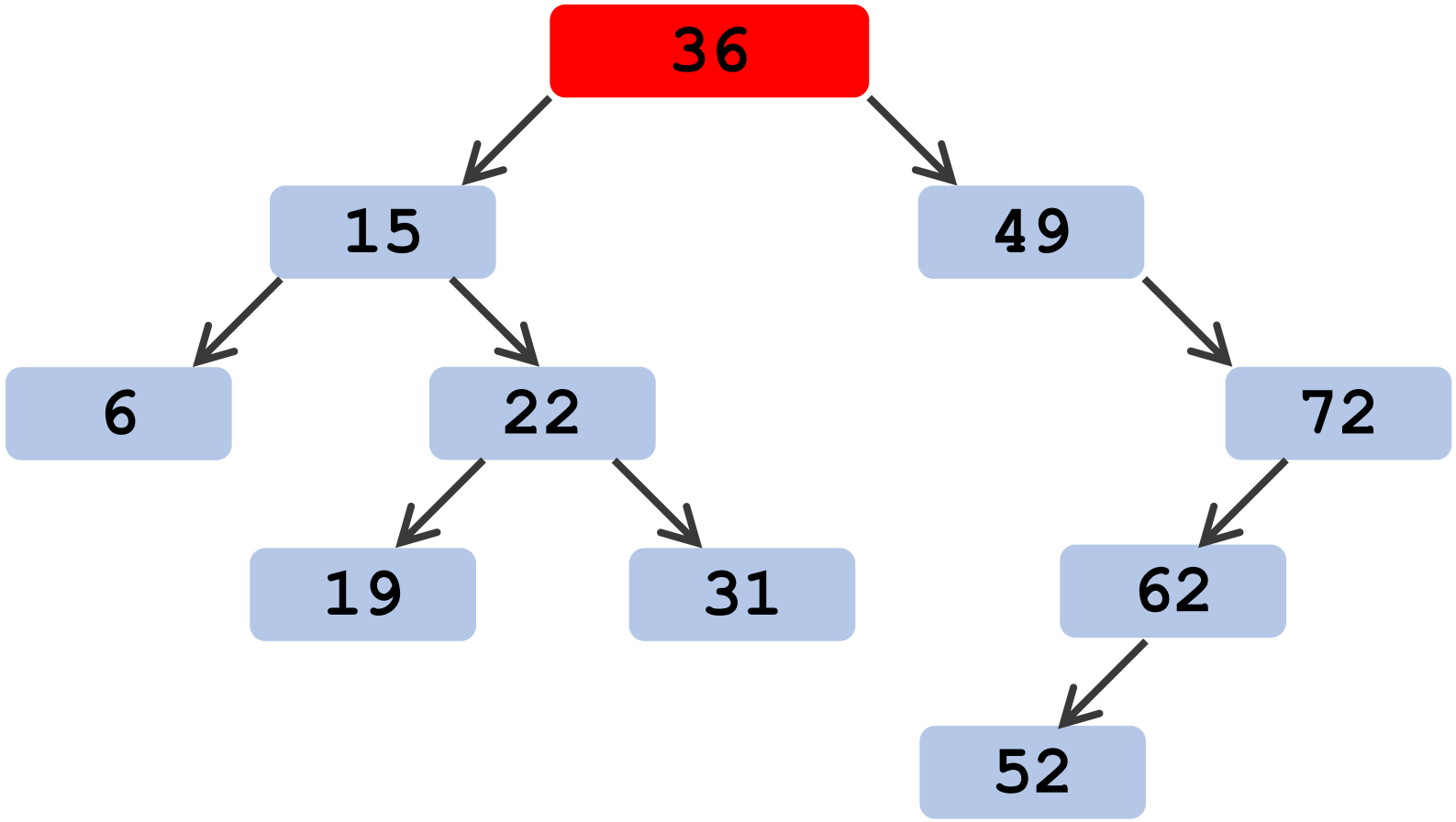
Find Successor

What is the successor of 36?



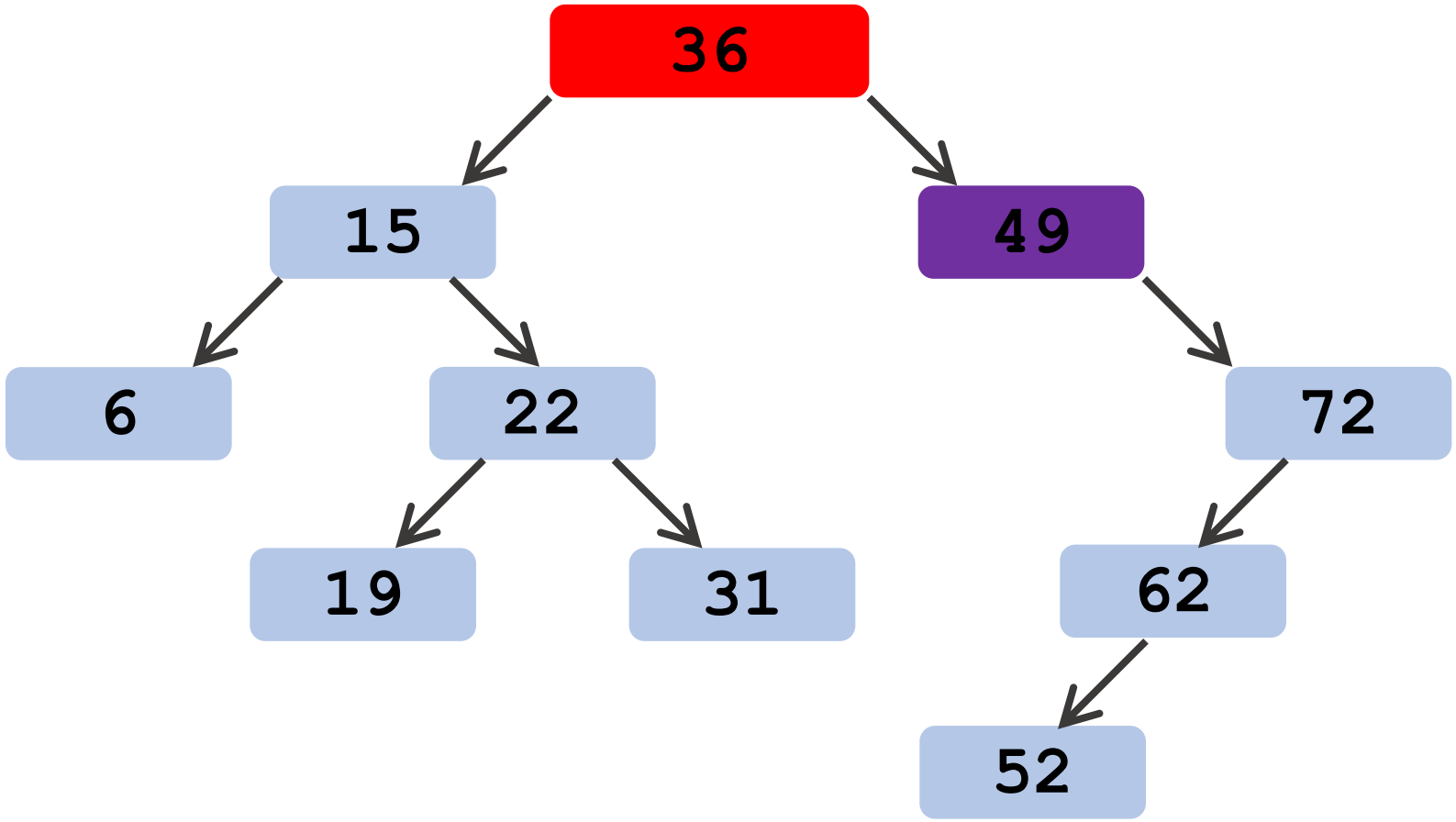
What is the successor of 36?

Inorder traversal: 6 15 19 22 31 36 49 52 62 72



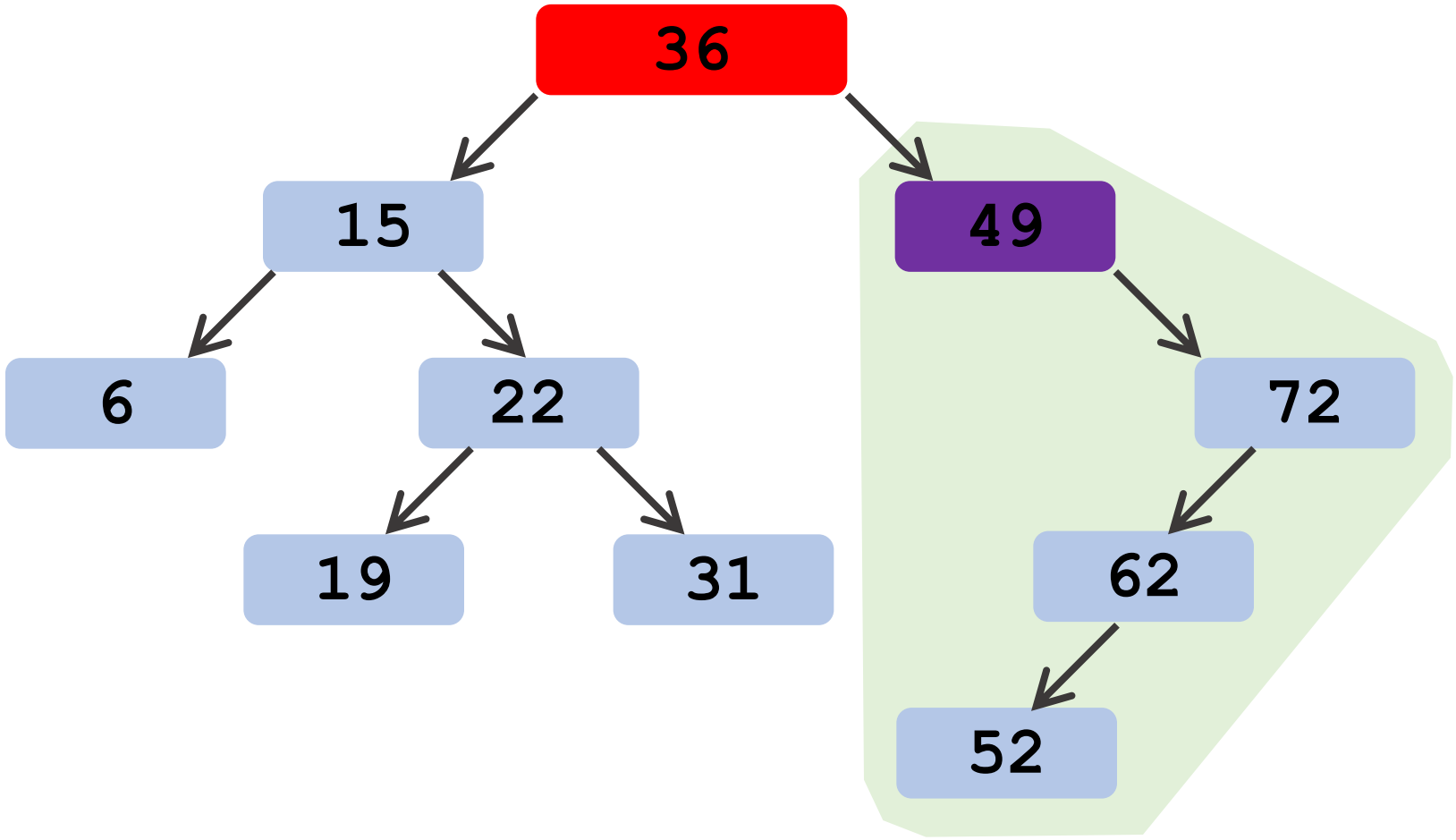
What is the successor of 36?

Inorder traversal: 6 15 19 22 31 36 49 52 62 72

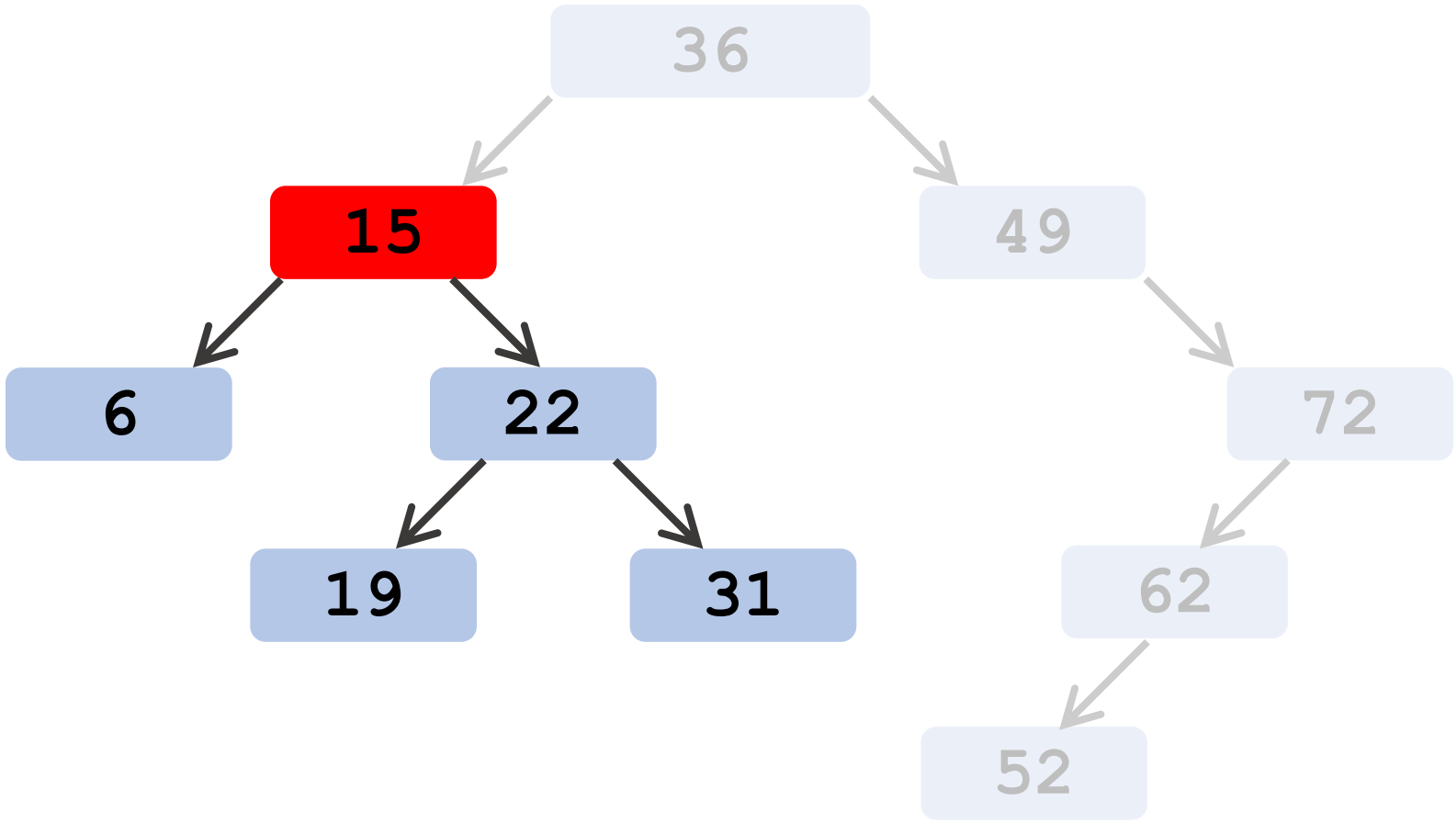


What is the successor of 36?

Successor is the leftmost vertex of the right sub-tree.

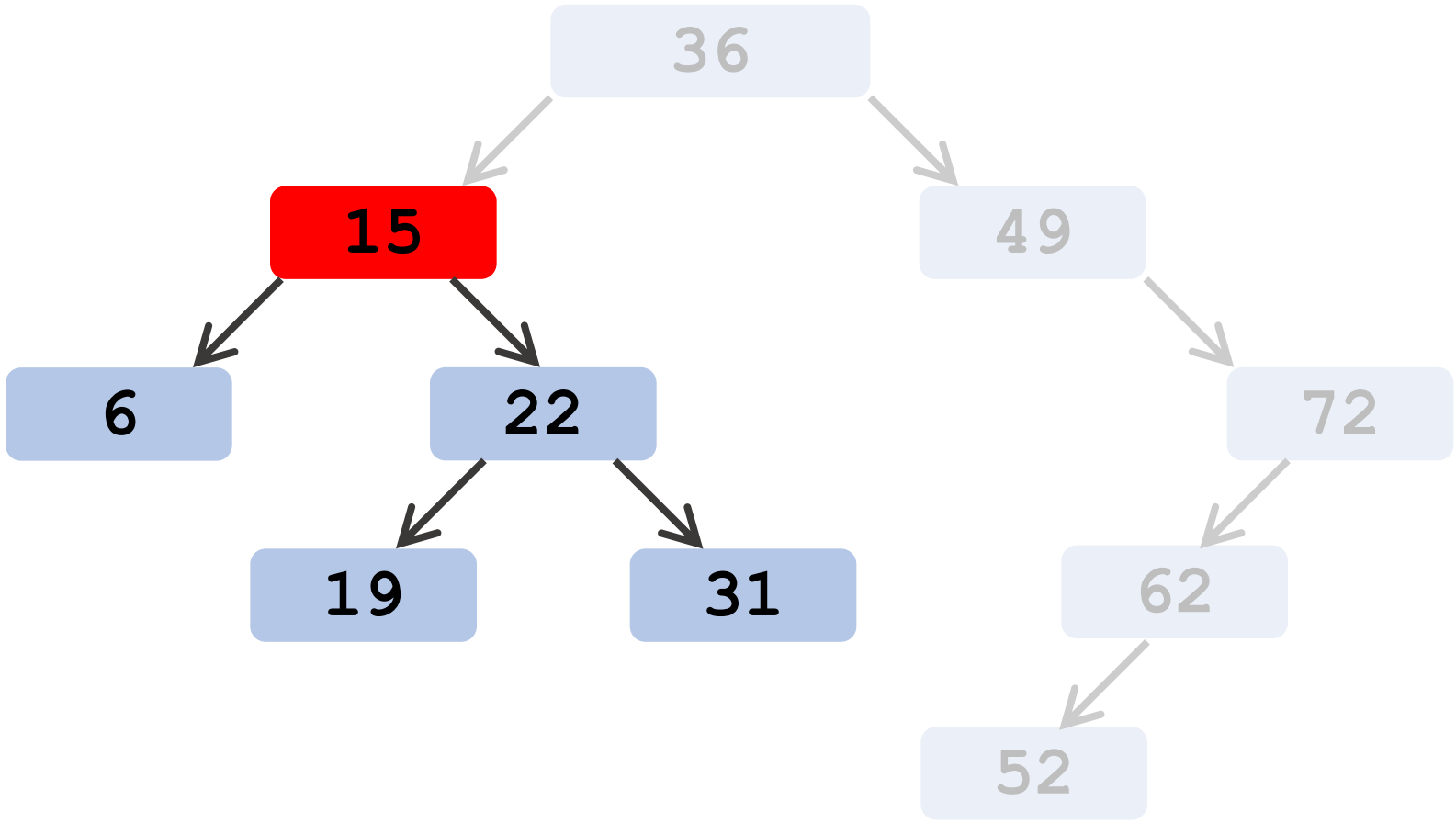


What is the successor of 15?



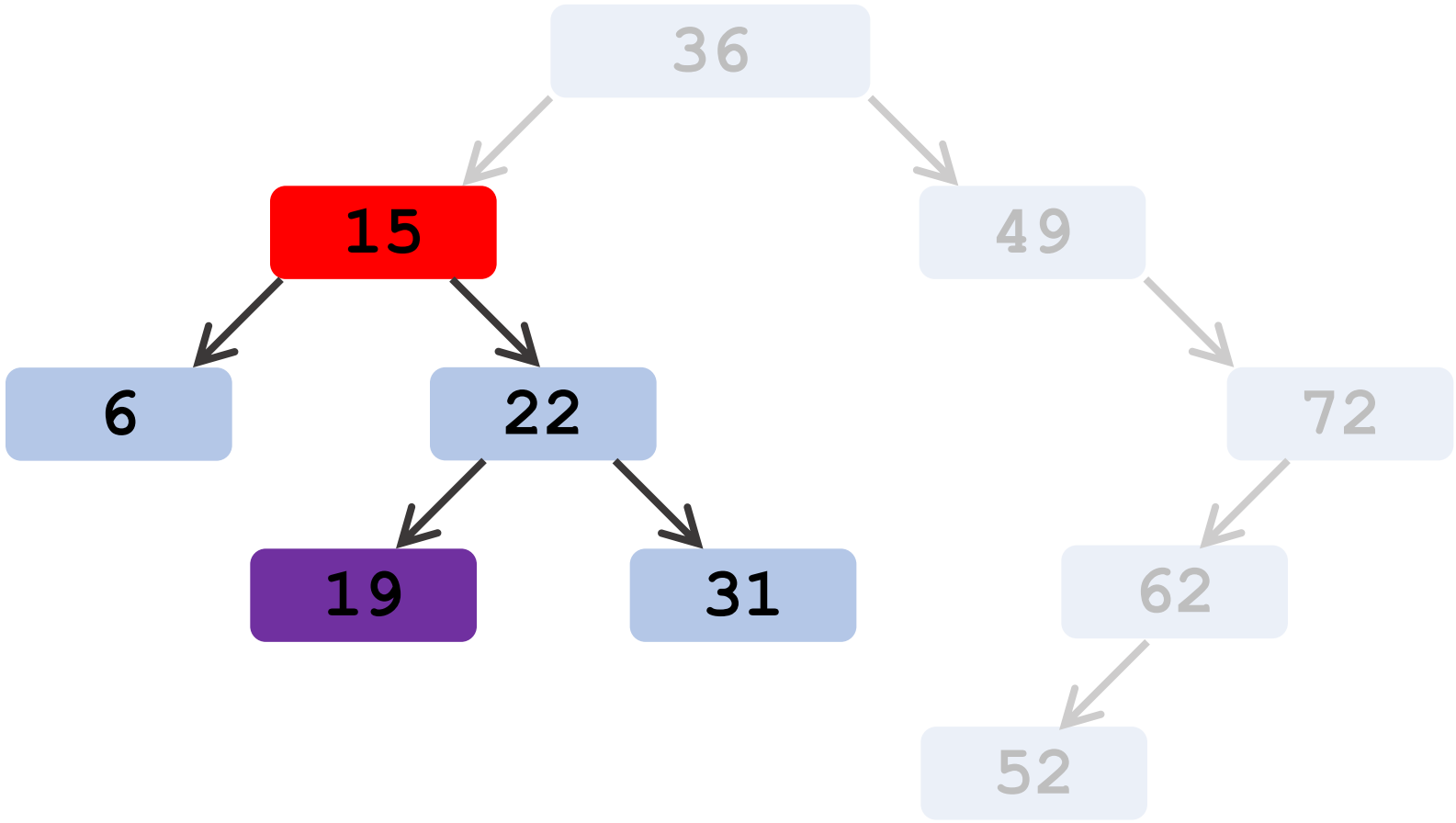
What is the successor of 15?

Inorder traversal: 6 | 15 | 19 | 22 | 31 | 36 | 49 | 52 | 62 | 72



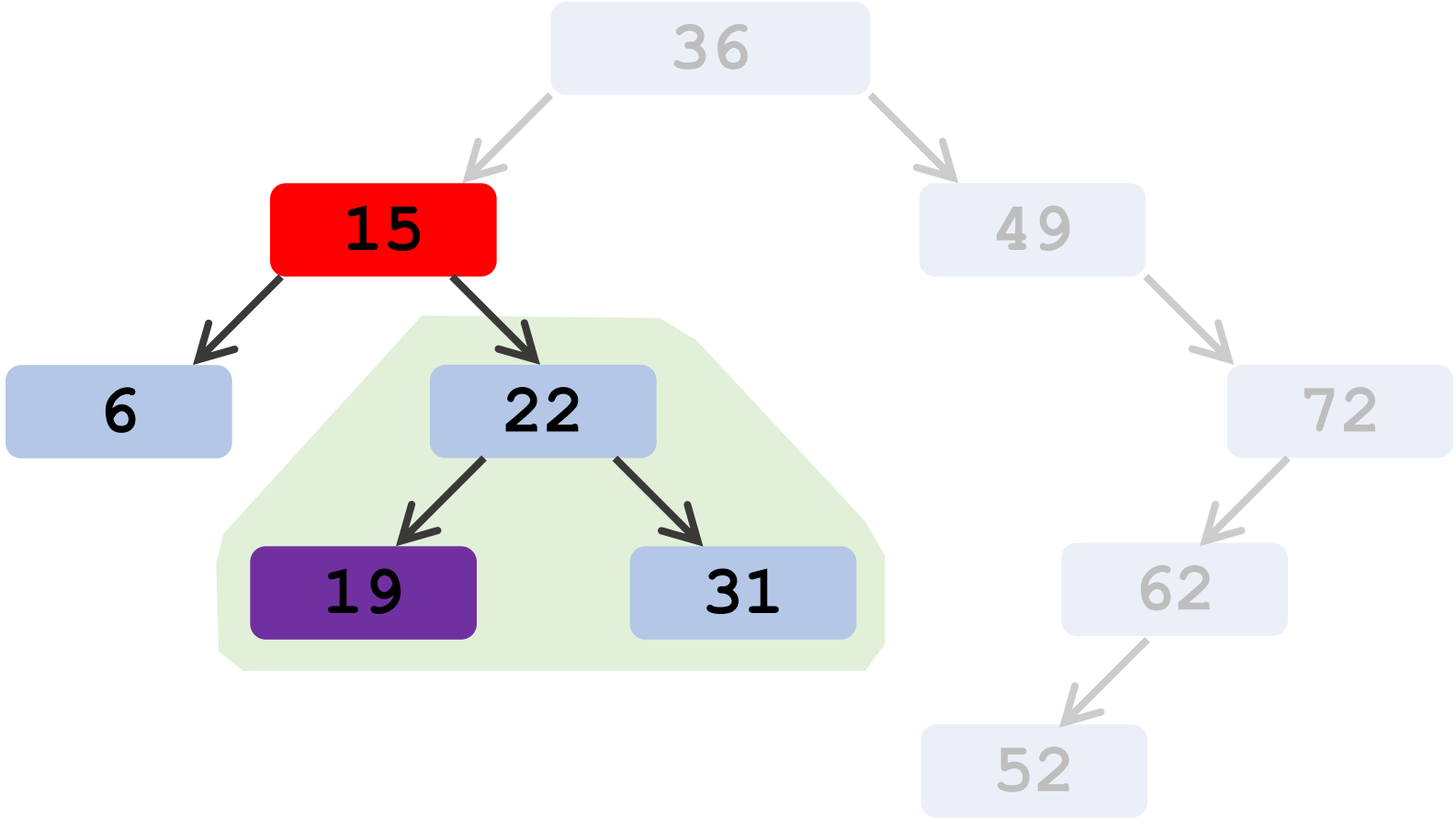
What is the successor of 15?

Inorder traversal: 6 | 15 | 19 | 22 | 31 | 36 | 49 | 52 | 62 | 72

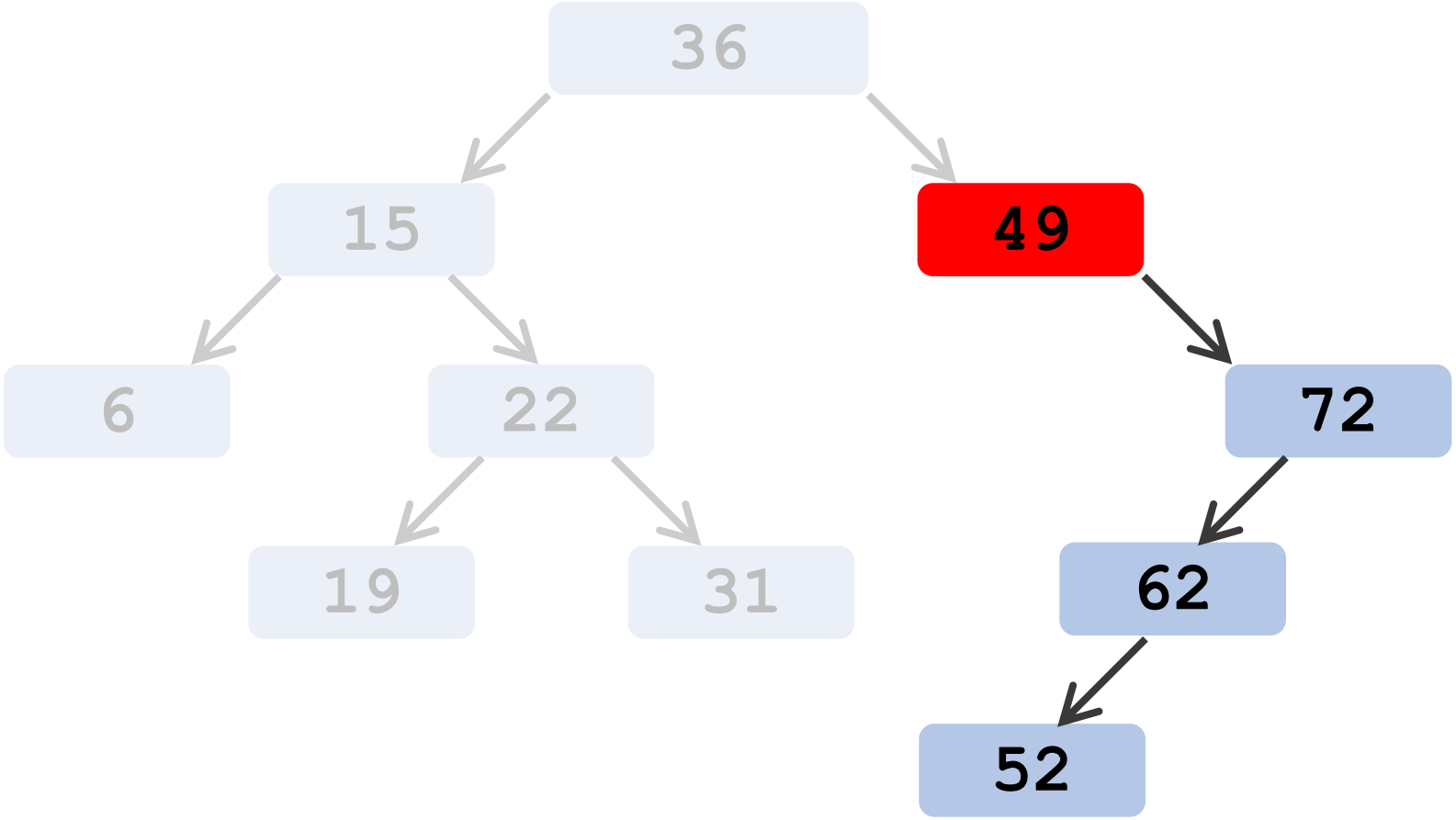


What is the successor of 15?

Successor is the leftmost vertex of the right sub-tree.

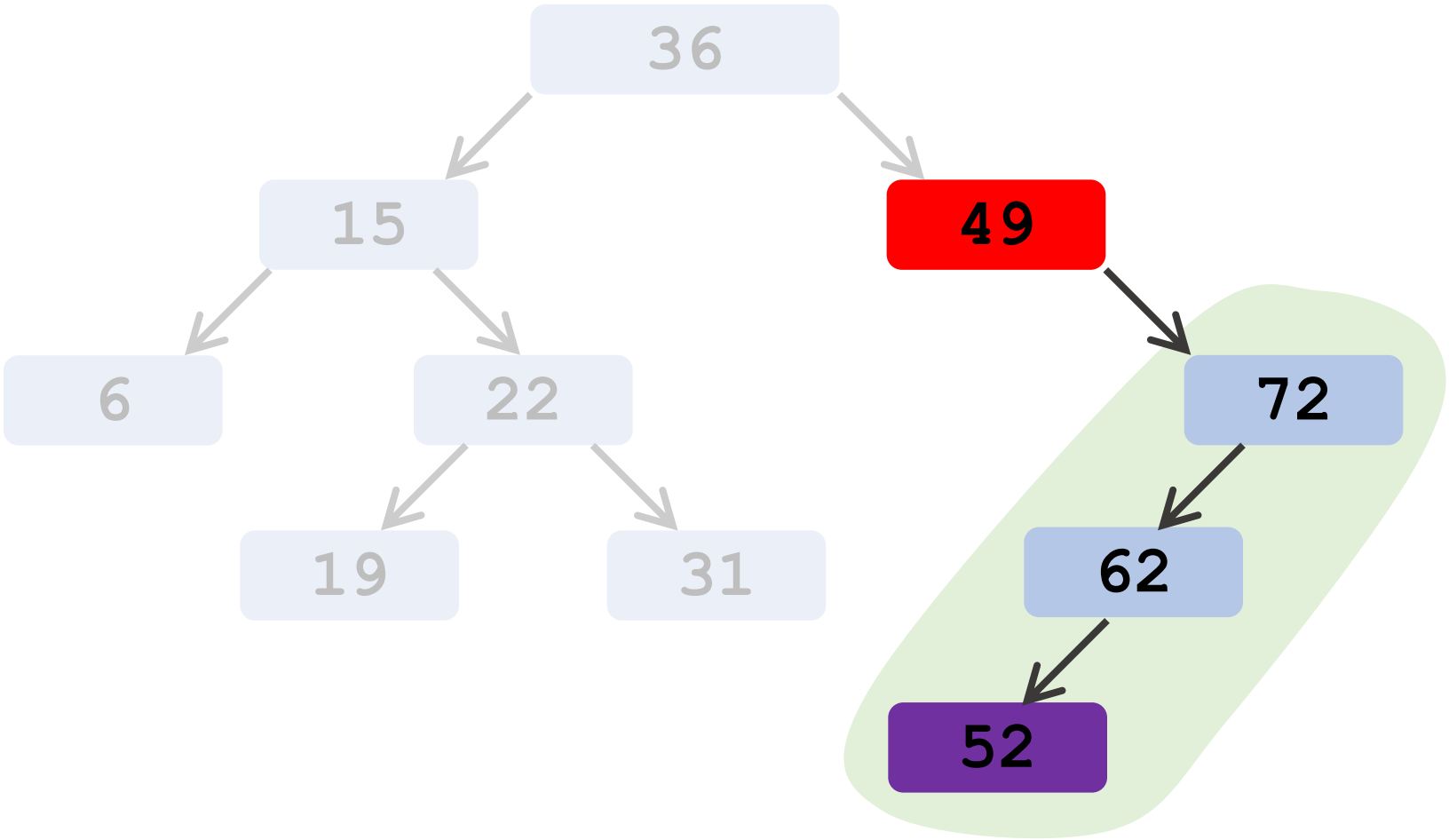


What is the successor of 49?

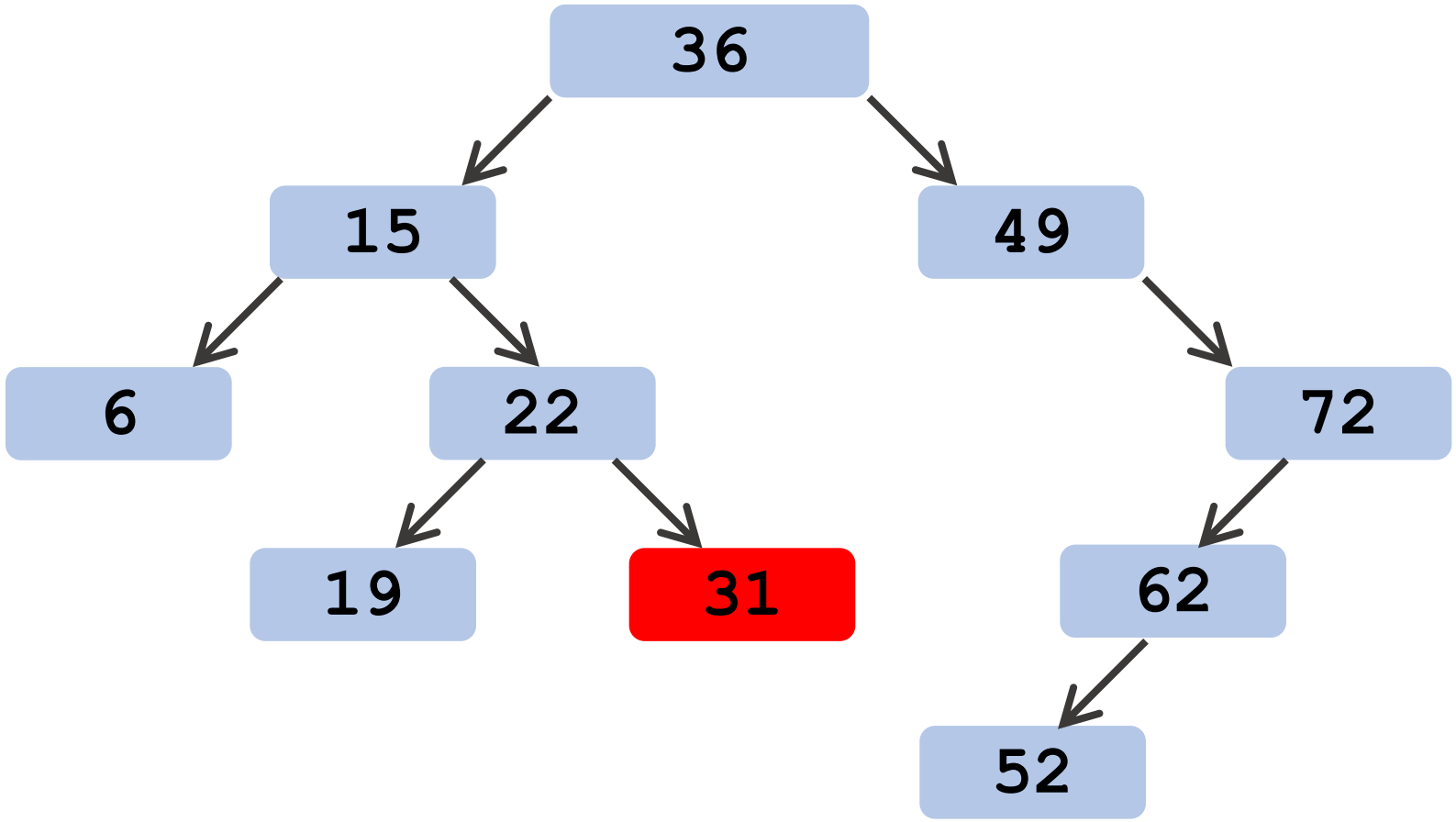


What is the successor of 49?

Successor is the leftmost vertex of the right sub-tree.

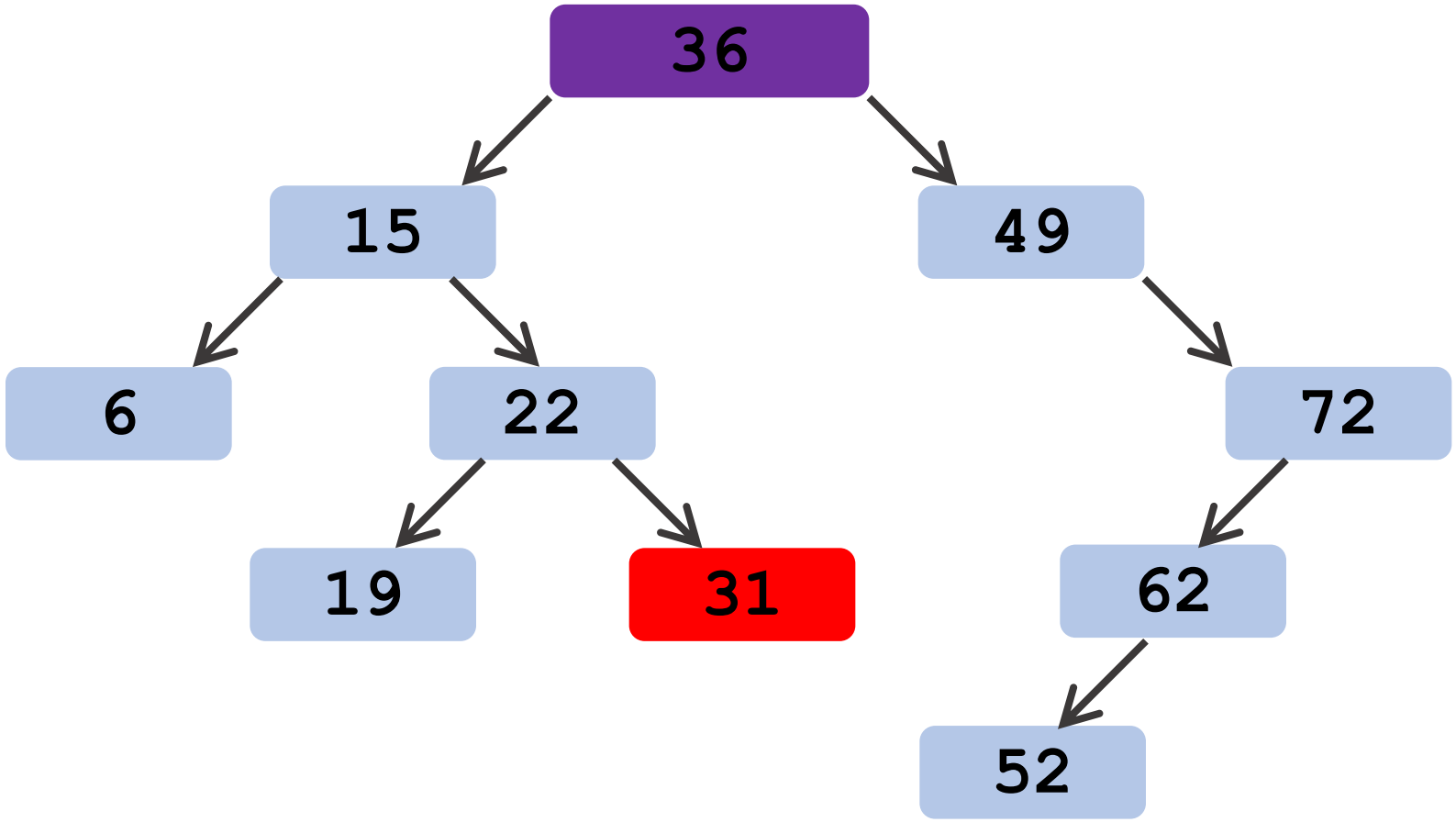


What is the successor of 31?



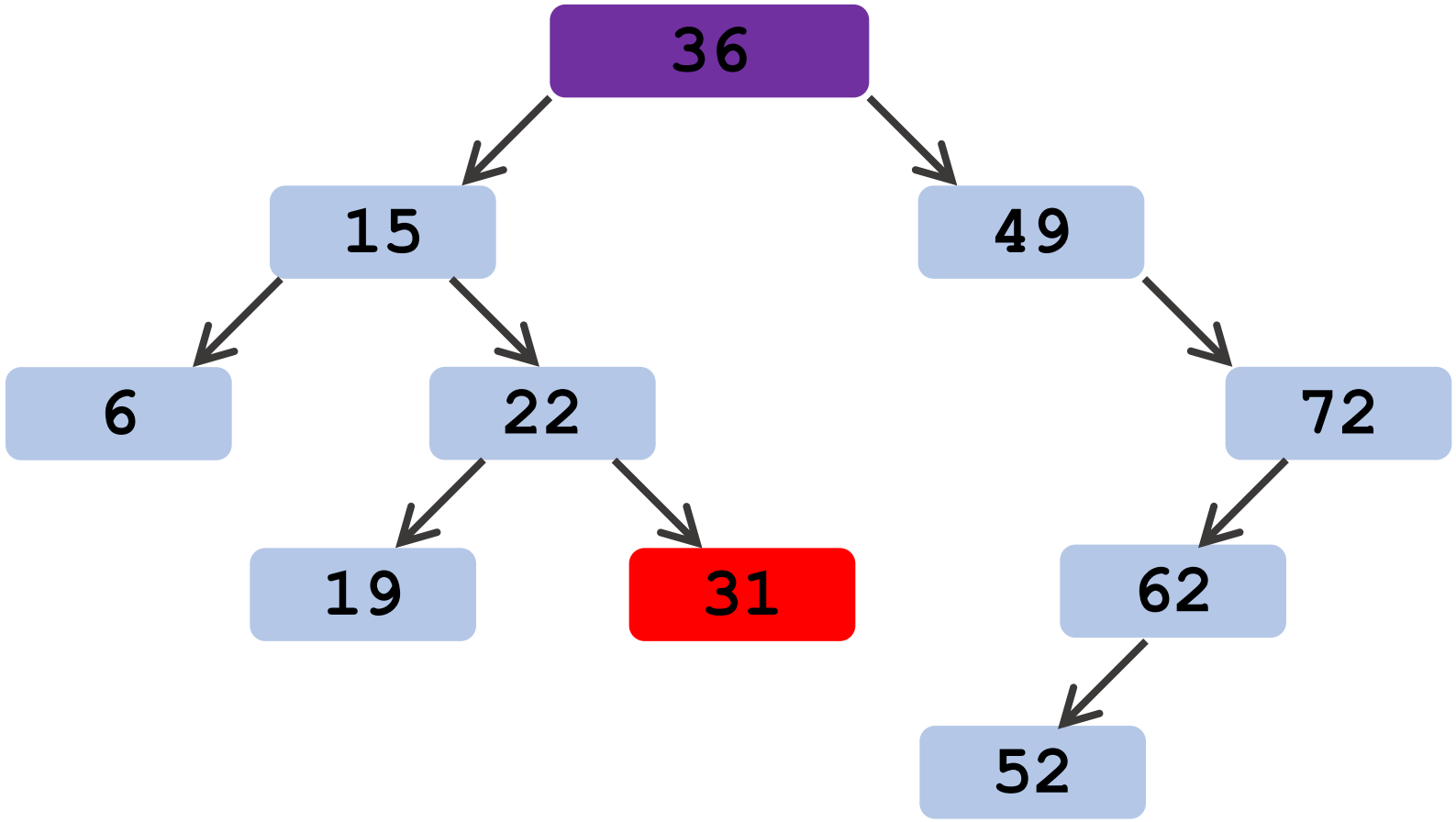
What is the successor of 31?

Inorder traversal: 6 15 19 22 31 36 49 52 62 72



What is the successor of 31?

Let's ignore such cases. Study vertices with right sub-tree.



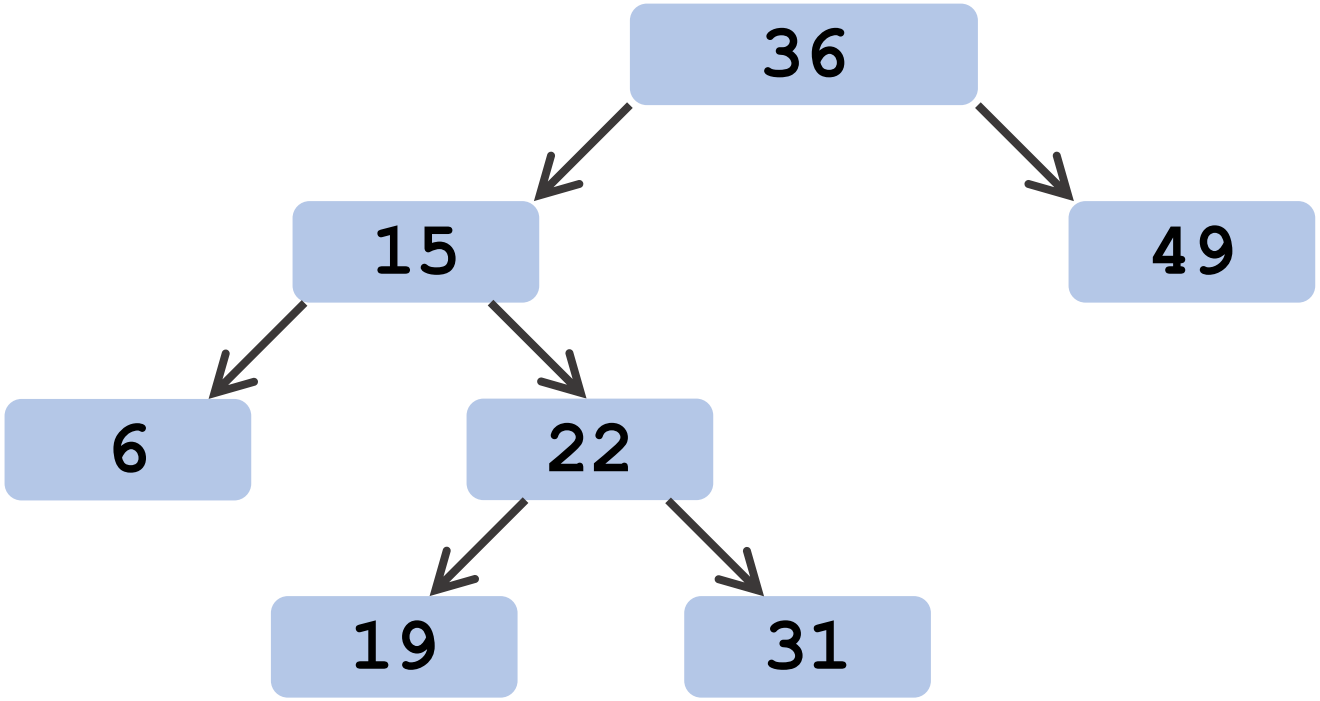
```
// find the leftmost vertex of a tree
struct vertex* leftmost(struct vertex* root) {
    struct vertex* current = root;
    while (current->left != NULL) {
        current = current->left;
    }
    return current;
}
```

```
// find the leftmost vertex of a tree
struct vertex* leftmost(struct vertex* root) {
    struct vertex* current = root;
    while (current->left != NULL) {
        current = current->left;
    }
    return current;
}
// assume vertex v has right child
// the successor of v
struct vertex* successor = leftmost(v->right);
```

Questions

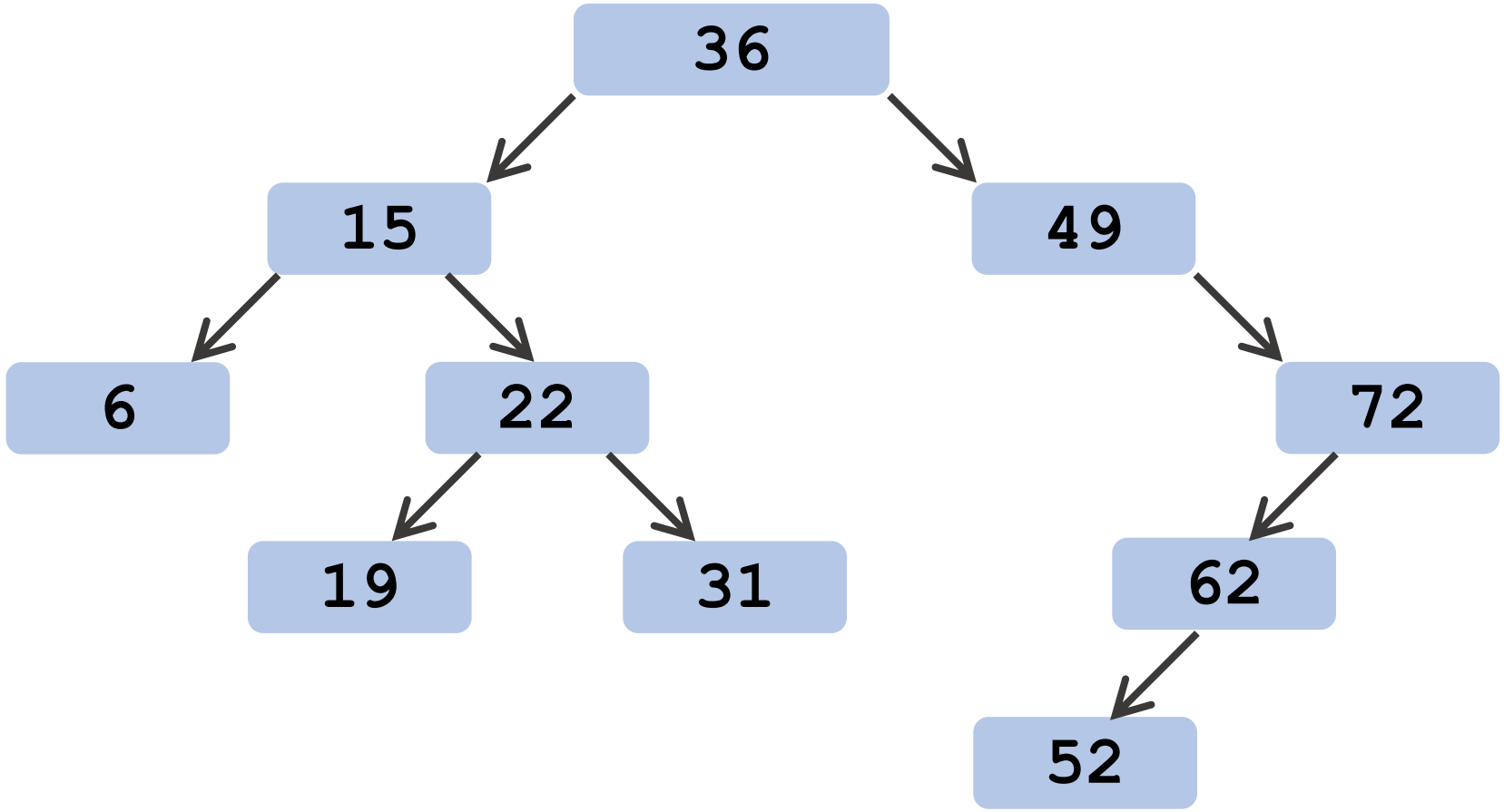
Question 1: Traversal

What are the results of inorder, preorder, and postorder print?



Question 2: Traversal

What are the results of inorder, preorder, and postorder print?



Question 3: Insertion

- Initially, the binary search tree is empty.
- The following keys are inserted sequentially:

`19, 89, 64, 8, 9, 6, 4, 66, 76.`

- Draw the tree after all the insertions.

Thank You!

http://wangshusen.github.io/