



链滴

# golang 实现二叉树

作者: [269812428](#)

原文链接: <https://ld246.com/article/1523624629464>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
package main

import (
    "fmt"
    "container/list"
)

type MyStack struct {
    List *list.List
}

type BinaryTree struct {
    Value interface{}
    Left *BinaryTree
    Right *BinaryTree
}

type Tree struct {
    Value interface{}
    Children []*Tree
}

func (stack *MyStack) pop() interface{} {
    if elem := stack.List.Back(); elem != nil {
        stack.List.Remove(elem)
        return elem.Value
    }
    return nil
}

func (stack *MyStack) push(elem interface{}) {
    stack.List.PushBack(elem)
}

func preOrderRecur(node *BinaryTree) {
    if node == nil {
        return
    }

    fmt.Println(node.Value)
    preOrderRecur(node.Left)
    preOrderRecur(node.Right)
}
```

```
}

func main() {
    node7 := &BinaryTree{Value: 7}
    node6 := &BinaryTree{Value: 6}
    node5 := &BinaryTree{Value: 5}
    node4 := &BinaryTree{Value: 4}
    node3 := &BinaryTree{Value: 3, Left: node6, Right: node7}
    node2 := &BinaryTree{Value: 2, Left: node4, Right: node5}
    root := &BinaryTree{Value: 1, Left: node2, Right: node3}
    preOrderRecur(root)
    fmt.Println()
    preOrder(root)
}

}
```