-

```
package main

import (
        "fmt"
        "github.com/gogf/gf/v2/container/gtree"
        "github.com/gogf/gf/v2/util/gutil"
)

func main() {
        m := gtree.NewRedBlackTree(gutil.ComparatorInt)

        //
        for i := 0; i < 10; i++ {
                m.Set(i, i*10)
        }
        //
        fmt.Println(m.Size())
        // ()
        m.Sets(map[interface{}]interface{}{
                10: 10,
                11: 11,
        })
        fmt.Println(m.Size())

        //
        fmt.Println(m.Contains(1))

        //
        fmt.Println(m.Get(1))

        //
        m.Remove(9)
        fmt.Println(m.Size())

        //
        m.Removes([]interface{}{10, 11})
        fmt.Println(m.Size())

        // ()
        fmt.Println(m.Keys())
        // ()
        fmt.Println(m.Values())

        //
        fmt.Println(m.GetOrSet(100, 100))

        //
        fmt.Println(m.Remove(100))

        // map
        m.IteratorAsc(func(k interface{}, v interface{}) bool {
                fmt.Printf("%v:%v ", k, v)
                return true
        })
        fmt.Println()

        // map
        m.Clear()

        // map
        fmt.Println(m.IsEmpty())
}
```

**Content Menu**

```
10
12
true
10
11
9
[0 1 2 3 4 5 6 7 8]
[0 10 20 30 40 50 60 70 80]
100
100
0:0 1:10 2:20 3:30 4:40 5:50 6:60 7:70 8:80
true
```

/

```
package main

import (
        "fmt"
        "github.com/gogf/gf/v2/container/gtree"
        "github.com/gogf/gf/v2/util/gutil"
)

func main() {
        tree := gtree.NewAVLTree(gutil.ComparatorInt)
        for i := 0; i < 10; i++ {
                tree.Set(i, i*10)
        }
        //
        tree.Print()
        //
        fmt.Println("ASC:")
        tree.IteratorAsc(func(key, value interface{}) bool {
                fmt.Println(key, value)
                return true
        })
        //
        fmt.Println("DESC:")
        tree.IteratorDesc(func(key, value interface{}) bool {
                fmt.Println(key, value)
                return true
        })
}
```

```
AVLTree
            9
          8
    7
            6
        5
              4
 3
          2
     1
           0

ASC:
0 0
1 10
2 20
3 30
4 40
5 50
6 60
7 70
8 80
9 90
DESC:
9 90
8 80
7 70
6 60
5 50
4 40
3 30
2 20
1 10
0 0
```