

# Golang 切片排序算法以及 sort 包

主讲教师：（大地）

合作网站：[www.itying.com](http://www.itying.com) （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

1、 选择排序.....	1
2、 冒泡排序.....	1
3、 Golang 内置 Sort 包对切片进行排序.....	2

## 1、选择排序

选择排序：进行从小到大排序

概念：通过比较，首先选出最小的数放在第一个位置上，然后在其余的数中选出次小数放在第二个位置上,依此类推,直到所有的数成为有序序列。

```
var numSlice = []int{9, 8, 7, 6, 5, 4}

for i := 0; i < len(numSlice); i++ {
    for j := i + 1; j < len(numSlice); j++ {
        if numSlice[i] > numSlice[j] {
            temp := numSlice[i]
            numSlice[i] = numSlice[j]
            numSlice[j] = temp
        }
    }
}

fmt.Println(numSlice)
```

## 2、冒泡排序

概念：从头到尾,比较相邻的两个元素的大小,如果符合交换条件,交换两个元素的位置。

特点：每一轮比较中,都会选出一个最大的数，放在正确的位置。

```
var numSlice = []int{9, 8, 7, 6, 5, 4}

for i := 0; i < len(numSlice); i++ {
    for j := i + 1; j < len(numSlice); j++ {
        if numSlice[i] > numSlice[j] {
            temp := numSlice[i]
            numSlice[i] = numSlice[j]
            numSlice[j] = temp
        }
    }
}

fmt.Println(numSlice)
```

### 3、Golang 内置 Sort 包对切片进行排序

#### 1、sort 包的文档：

<http://docsen.studygolang.com/pkg/sort/>

<https://golang.org/src/sort>

#### 2、sort 升序排序

对于 int 、 float64 和 string 数组或是切片的排序， go 分别提供了 sort.Ints() 、 sort.Float64s() 和 sort.Strings() 函数， 默认都是从小到大排序。

```
intList := []int{2, 4, 3, 5, 7, 6, 9, 8, 1, 0}

float8List := []float64{4.2, 5.9, 12.4, 10.2, 50.7, 99.9, 31.4, 27.81828, 3.14}

stringList := []string{"a", "c", "b", "z", "x", "w", "y", "d", "f", "i"}

sort.Ints(intList)
```

```
sort.Float64s(float8List)
```

```
sort.Strings(stringList)
```

输出:

```
[0 1 2 3 4 5 6 7 8 9]
```

```
[3.14 4.2 5.9 10.2 12.4 27.81828 31.4 50.7 99.9]
```

```
[a b c d f i w x y z]
```

### 3、sort 降序排序

Golang 的 sort 包可以使用 sort.Reverse(slice) 来调换 slice.Interface.Less，也就是比较函数，所以，int、float64 和 string 的逆序排序函数可以这么写。

```
intList := []int{2, 4, 3, 5, 7, 6, 9, 8, 1, 0}
```

```
float8List := []float64{4.2, 5.9, 12.4, 10.2, 50.7, 99.9, 31.4, 27.81828, 3.14}
```

```
stringList := []string{"a", "c", "b", "z", "x", "w", "y", "d", "f", "i"}
```

```
sort.Sort(sort.Reverse(sort.IntSlice(intList)))
```

```
sort.Sort(sort.Reverse(sort.Float64Slice(float8List)))
```

```
sort.Sort(sort.Reverse(sort.StringSlice(stringList)))
```

```
fmt.Printf("%v\n%v\n%v\n", intList, float8List, stringList)
```

```
[9 8 7 6 5 4 3 2 1 0]
```

```
[99.9 50.7 31.4 27.81828 12.4 10.2 5.9 4.2 3.14]
```

```
[z y x w i f d c b a]
```