链滴

golang 之几种常见排序算法

作者：xhaoxiong

原文链接：https://ld246.com/article/1525342820794

来源网站：链滴

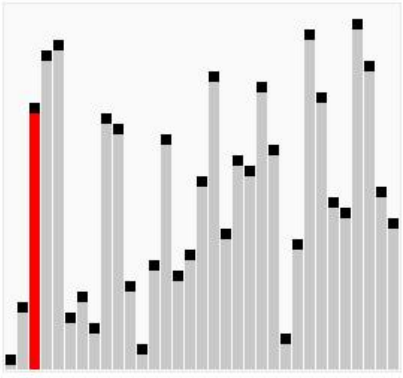
```
package main


import "fmt"

func main() {
    //values := []int{88, 66, 44, 45, 98, 74, 1, 8, 102, 99, 74, 6}
    values := []int{9, 0, 6, 5, 8, 2, 1, 7, 4, 3}
    fmt.Println(values)
    //Popsort(values)
    //Insertsort(values)
    //Selectsort(values)
    fmt.Println(values)
}

// 平均:O(n^2) 好:O(n) 坏:O(n^2) 空间:O(1) 稳定性:稳定
func Popsort(values []int) {
    for i := 0; i < len(values)-1; i++ {
        for j := i + 1; j < len(values); j++ {
            if values[i] > values[j] { //当前的数逐一与后面的数比较，如果大则交换位置
                //A->Z
                values[i], values[j] = values[j], values[i]
            }
        }
    }
}
```

6 5 3 1 8 7 2 4



```
// 平均:O(n^2) 好:O(n) 坏:O(n^2) 空间:O(1) 稳定性:稳定
func Insertsort(values []int) {
    for i := 1; i < len(values); i++ { // 类似抓扑克牌排序
        get := values[i] // 右手抓到一张扑克牌
        j := i - 1       // 拿在左手上的牌总是排序好的
```

```
        for j >= 0 && values[j] > get { // 将抓到的牌与手牌从右向左进行比较
            values[j+1] = values[j] // 如果该手牌比抓到的牌大，就将其右移
            j--
        }
        values[j+1] = get //就是将get这个数放到通过循环减减的j下标+1位置,直到该手牌比抓到的牌(或二者相等)，将抓到的牌插入到该手牌右边(相等元素的相对次序未变，所以插入排序是稳定的)
    }
}
```

6 5 3 1 8 7 2 4

```
// 平均:O(n^2) 好:O(n^2) 坏:O(n^2) 空间:O(1) 稳定性:不稳定
func Selectsort(values []int) {

    for i := 0; i < len(values)-1; i++ {
        min := i

        for j := i + 1; j < len(values); j++ {
            if values[j] < values[min] {
                min = j
            }
        }
        if min != i {
            values[min], values[i] =
                values[i], values[min]
        }

    }
}
```

	8
	5
	2
	6
	9
	3
	1
	4
	0
	7

各种常用排序算法						
类别	排序方法	时间复杂度			空间复杂度	稳定性
		平均情况	最好情况	最坏情况	辅助存储	
插入排序	直接插入	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	稳定
	shell排序	$O(n^{1.3})$	$O(n)$	$O(n^2)$	$O(1)$	不稳定
选择排序	直接选择	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
	堆排序	$O(nlog_2n)$	$O(nlog_2n)$	$O(nlog_2n)$	$O(1)$	不稳定
交换排序	冒泡排序	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	稳定
	快速排序	$O(nlog_2n)$	$O(nlog_2n)$	$O(n^2)$	$O(nlog_2n)$	不稳定
归并排序		$O(nlog_2n)$	$O(nlog_2n)$	$O(nlog_2n)$	$O(1)$	稳定
基数排序		$O(d(r+n))$	$O(d(n+rd))$	$O(d(r+n))$	$O(rd+n)$	稳定
注：基数排序的复杂度中，r代表关键字的基数，d代表长度，n代表关键字的个数						

原文链接：golang 之几种常见排序算法