中国科学院大学
University of Chinese Academy of Sciences

计算机科学导论实验

编程基础-3

简版快速排序程序fastsort.go

函数签名

zxu@ict.ac.cn
zhangjialin@ict.ac.cn

编程基础实验安排

<table>
<tr><th>周次</th><th>日期</th><th>实验课</th><th>课程内容</th><th>作业提交内容</th><th>截止时间</th></tr>
<tr><td>2</td><td>3月8日</td><td>编程基础-1</td><td rowspan="2">从hello到Name2Number.go
基本数据类型、基本语句、循环</td><td>name_to_number-0.go文件与程序结果</td><td rowspan="4">2024/3/24
23:30</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">3月15日</td><td rowspan="2">编程基础-2</td><td>Name2Number.go文件与程序结果</td></tr>
<tr><td rowspan="2">切片与递归
简版快速排序程序
fastsort.go</td><td>fastsort.go文件与程序中间结果</td></tr>
<tr><td>4</td><td>3月22日</td><td>编程基础-3</td><td>修改函数签名后的fastsort.go文件与程序中间结果</td></tr>
</table>

- 作业提交后会立刻返回成绩。
- 在截止时间前四次作业均可反复多次提交。

提纲

- 目标：
  - 修改fastsort.go排序程序，理解递归与切片

- 步骤
  - 修改fastsort.go程序，修改partition函数签名
  - 按照程序手工模拟程序运行结果

课件中包含教科书未包括的素材引用，特此致谢

1. 函数签名

- 程序中的函数很像数学函数
  - 输入数据→输出数据

参数列表，告诉函数在运行时需要
接收什么样的信息
此处 A 是一个整数切片

返回值的类型，告诉函数在运行
结束后会返回什么样的结果
此处函数会返回两个整数切片

```
func partition(A []int) ([]int, []int) {  // len(A)>1
    lower := 0  // lower 是lowerA的长度，初始值为0
    for i:= 0; i<len(A); i++ {  // 逐个扫描A的元素A[i]
        // 此处插入你的代码
        // 建议：如果A[i]<标杆值，
        //  A[i]与A[lower]交换并更新lower
    }
    // 此时lower的值是len(lowerA)+1;
    // 交换，使得标杆紧跟在lowerA之后
    A[lower], A[len(A)-1] = A[len(A)-1], A[lower]
    // 返回切片lowerA和upperA
    // 注意两个切片的起始索引和结束索引
    return A[0:lower], A[lower+1:len(A)]
}
```

2. 修改fastsort.go

- 实验要求：
  - 利用右侧代码框架
    - 修改partition函数签名，在partition函数的返回值中增加pivot，使partition函数按序依次返回lowerA, pivot, upperA三个变量
    - 在fastsort函数中按序依次打印lowerA, pivot, upperA

可以任意修改partition函数体

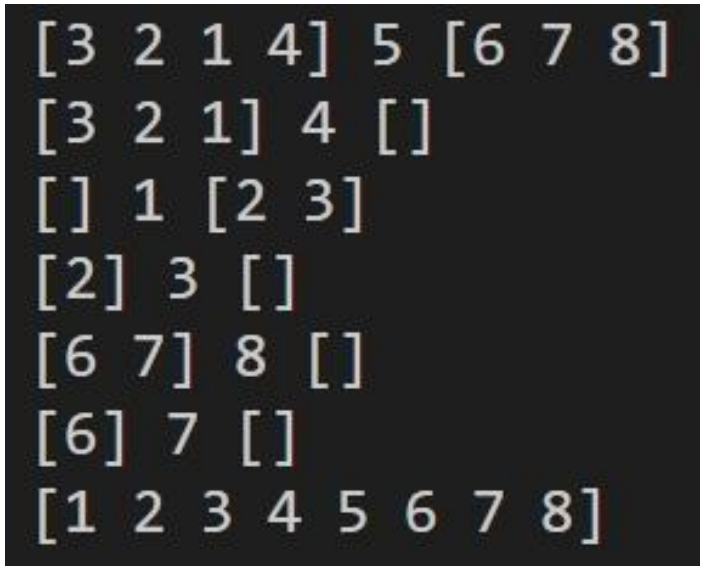
```
package main  // fastsort.go
import "fmt"
var d [8]int = [8]int {8,3,6,7,2,1,4,5}
var s []int = d[0:8]
func main() {
   fastsort(s)
   fmt.Println(s)
}

func fastsort(A []int) {
   if len(A) < 2 {  // 判断是不是基线情况
       return       // 是，退出
   }
   lowerA, upperA := partition(A)  // 修改此处
   // 此处插入你的代码
   // 打印lowerA, pivot, upperA
   fastsort(lowerA) // 递归排序小数组
   fastsort(upperA) // 递归排序小数组
}

func partition(A []int) ([]int, []int) {  // 修改此处
   lower := 0
   for i:= 0; i<len(A); i++ {
      // 此处插入上次课时你的代码
      // 建议：如果A[i]<标杆值，
      //  A[i]与A[lower]交换并更新lower
   }
   A[lower], A[len(A)-1] = A[len(A)-1], A[lower]
   return A[0:lower], A[lower+1:len(A)]  // 修改此处
}
```

2. 修改fastsort.go

- 样例输出




```
package main  // fastsort.go
import "fmt"
var d [8]int = [8]int {8,3,6,7,2,1,4,5}
var s []int = d[0:8]
func main() {
   fastsort(s)
   fmt.Println(s)
}

func fastsort(A []int) {
   if len(A) < 2 {  // 判断是不是基线情况
       return       // 是，退出
   }
   lowerA, upperA := partition(A)  // 修改此处
   // 此处插入你的代码
   // 打印lowerA, pivot, upperA
   fastsort(lowerA) // 递归排序小数组
   fastsort(upperA) // 递归排序小数组
}

func partition(A []int) ([]int, []int) {  // 修改此处
   lower := 0
   for i:= 0; i<len(A); i++ {
      // 此处插入上次课时你的代码
      // 建议：如果A[i]<标杆值，
      //  A[i]与A[lower]交换并更新lower
   }
   A[lower], A[len(A)-1] = A[len(A)-1], A[lower]
   return A[0:lower], A[lower+1:len(A)]  // 修改此处
}
```

3. 提交fastsort.go

- 实验要求：
  - 提交修改后的fastsort.go文件
  - 提交lowerA, pivot, upperA的值
    - 由于采用计算机自动评分，请严格按照云平台示例填写内容

作业提交链接
https://course.things.ac.cn:10088/exp/basic_programming

作业提交截止时间
2024/3/24 23:30

请提交

请把fastsort.go中的待排序数组设成如下值: [2, 5, 3, 4, 1, 9]

- fastsort.go 选择文件 未选择任何文件

请使用给定的数组，填写fastsort函数中按partition函数执行顺序依次打印的lowerA, pivot, upperA。由于采用计算机自动评分，请**严格**按如下示例填写内容

假如第1次打印的结果是lower = [2,3,5], pivot=6, upper=[7]，则在第1行的第1个空格填写2,3,5 （数组各数字之间使用1个英文逗号隔开）第2个空格填写6 第3个空格填写7

假如第2次打印的结果是lower = [2,3], pivot=5, upper=[]，则在第2行的第1个空格填写2,3 第2个空格填写5 第3个空格不填写任何内容

假如partition仅执行了3次或4次，则只填前3行或前4行，其它行数不需要填写任何内容

	Lower	pivot	Upper
1:			
2:			
3:			
4:			
5:			

提交

4. 阅读教科书

- 认真阅读中文教科书2.2节与2.3节（P55-85）
  - 复现书中Go语言例子
  - 学习斐波那契计算机抽象