**江西师范大学计算机信息工程学院学生实验报告（5）**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **专业：** | **数据科学与大数据技术2班** | **姓名：** | **赖丽婷** | **学号：** |  | **日期：** | **2021.10.24** |

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 数据结构 | 实验室名称 | 计算机综合实验室 |
| 实验名称 | 递归 | | |
| 指导教师 |  | 成绩 |  |

**1.实验目的**（结出本次实验所涉及并要求掌握的知识点）

1.1掌握递归的基本概念与递归程序设计

1.2能对程序执行过程进行分析

1.3掌握怎么编写递归程序找到递归出口

**2.实验内容**（结出实验内容具体描述）

2.1编写递归算法，求数组a[left..right]的最大值

2.2编写一个递归算法函数，将数组a[left..right]中的所有奇数调整到数组左边，所有偶数调整到数组右边

2.3编写递归函数将长度为n的数组采用冒泡进行升序排序，编写递归函数，采用二分查找值为key的元素所在的位置，查找失败函数返回-1

2.4编写程序返回带头单链表最大值所在的节点地址

**3.算法描述及实验步骤**（用适当的形式表达算法设计思想与算法实现步骤）

3.1第一题找数组最大值，递归出口在数组末尾，将当前函数的a[left]值与返回值进行比较如果当前函数的a[left]值大于返回值，则返回a[left]如果小于则返回下一层函数的返回值，直到所以函数执行完成就能返回最大数了

3.2第二题在数组左边找到偶数，然后在数组右边找到奇数，将他们互换，然后递归，最后当left>right时递归结束

3.3利用长度n进行冒泡排序，每排完一次递归N-1，可以设置一个boolean的值来判断是否还需要继续排序下去，没有进行换位置说明序列已经排好序了，可以退出递归，提高效率，二分法求key值的话，将key与中间位的数进行比较，如果比它大则left=mid如果比它小则right=mid以此递归，直到找到a[mid]=key返回mid

3.4对单链表进行访问，递归出口有两种情况，一个是当只有一个头节点时，直接返回NULL，一个是head->next->next=NULL时返回，不断嵌套，根据返回的值羽当前函数的head->next的值进行比较，返回更大的那个数。

**4.调试过程及运行结果**（详细记录在调试过程中出现的问题及解决方法。记录实验执行的结果）

**5. 总结**（对实验结果进行分析，问题回答，实验心得体会及改进意见）

要找准递归出口，了解递归程序的优势，也要知道递归的坏处，容易陷入死循环，空间复杂度也比较高，但是代码简洁，可以解决很多很多复杂的问题，根据传递参数的不同，程序设计也是不同的，

1. **附录**（程序源代码等）

6.1#include "ArrayIo.h"

#define N 10

/\*请将本函数补充完整，并进行测试\*/

//普通方法递归出口在最后面 然后让后面的跟前面的进行比较

int max(int a[],int left,int right)

{

int maxa, maxb;

maxb = a[left];

if(left < right)

{

maxa = max(a, left+1, right);

return maxa>maxb ? maxa : maxb;

}else{

return a[right];

}

}

6.2#include "ArrayIo.h"

//将奇数调整到左边，偶数调整到右边

void partion(int a[], int left, int right)

{

int mid;

if(left < right)

{

while(left < right && a[left]%2 != 0)

{

left++;

}

while(left < right && a[right]%2 ==0)

{

right--;

}

if(left < right)

{

mid = a[right];

a[right] = a[left];

a[left] = mid;

partion(a, left+1, right-1);

}

}

}

6.3#include "ArrayIo.h"

#define N 10

void bubbleSort(int a[], int n)

{

int i = 0,j;

if(n > 0)

{

for(i; i < n-1; i++)

{

if(a[i] > a[i+1])

{

j = a[i];

a[i] = a[i+1];

a[i+1] = j;

}

}

bubbleSort(a, n-1);

}

}

int binSearch(int a[], int left, int right, int key)

{

int mid;

if(left <= right)

{

mid = (left + right)/2;

if(a[mid] == key)

{

return mid;

}

else if(a[mid] > key)

{

binSearch(a, left, mid, key);

}else

{

binSearch(a, mid, right, key);

}

}else

{

return -1;

}

}

6.4#include "slnklist.h"

//因为是带头节点 可能只有一个头节点，或者有一个及以上节点 要对递归出口进行分类

//因为我们要对下一个节点进行访问所以要返回一个不为空的节点，如果head->next->next为空的话 返回head->next访问NULL->info会造成内存泄漏

linklist max(linklist head)

{

linklist m;

if(!head->next)

{

return NULL;

}

else if(!head->next->next)

{

return head->next;

}else

{

m = max(head->next);

return head->next->info > m->info ? head->next : m;

}

}