程序设计实践（一）

哈尔滨工业大学 计算机学院
任课教师：孙大烈教授
助教：付万增

分治

一、学习递归
二、归并排序
三、快速排序
四、习题解答

心得分享

回忆程序设计的学习过程

- 学习打印Holle，World！继而学会打印任何字符串；
- 学习四则运算，继而学会求f(x)的值；
- 学习递推问题，继而学会数组和循环；
- 学习素数判断，继而熟悉循环和条件结构；
- 学习背包问题，继而了解贪心思想；
- 学习二分查找，继而了解分治思想。

习题切入 → 高级模仿 → 课下训练

递归是我学习过程的一道坎，但不希望是你们的坎。

子程序

如何将程序包装起来？

- 子程序
- int max(int x,int y) //返回值类型声明 子程序名（参数）
- {
- 	int mx;
- 	if(x>y) mx = x;
- 	else mx = y;
- 	return 0; //返回值
- }
- int x = max(3,4); //调用子程序 x=4

递归

在子程序中调用子程序会发生什么？

- 子程序
- int func(int x) //返回值类型声明 子程序名（参数）
- {

```
    if(x == 0) return 1;
    If(x == 1) return 1;
    return func(x-1) + func(x-2);
```

- }
- int x = func(3); //调用子程序
- **//比较一下上述代码和之前的递推代码，比较效果和过程**

递归

什么是递归？

- 想象你拿着一面镜子站在另一面镜子面前
- 想象你在大峡谷里喊一声“Hello，World！”
- 程序调用自身的编程技巧称为递归（ recursion）。
- 一般来说，递归需要有边界条件。

递归

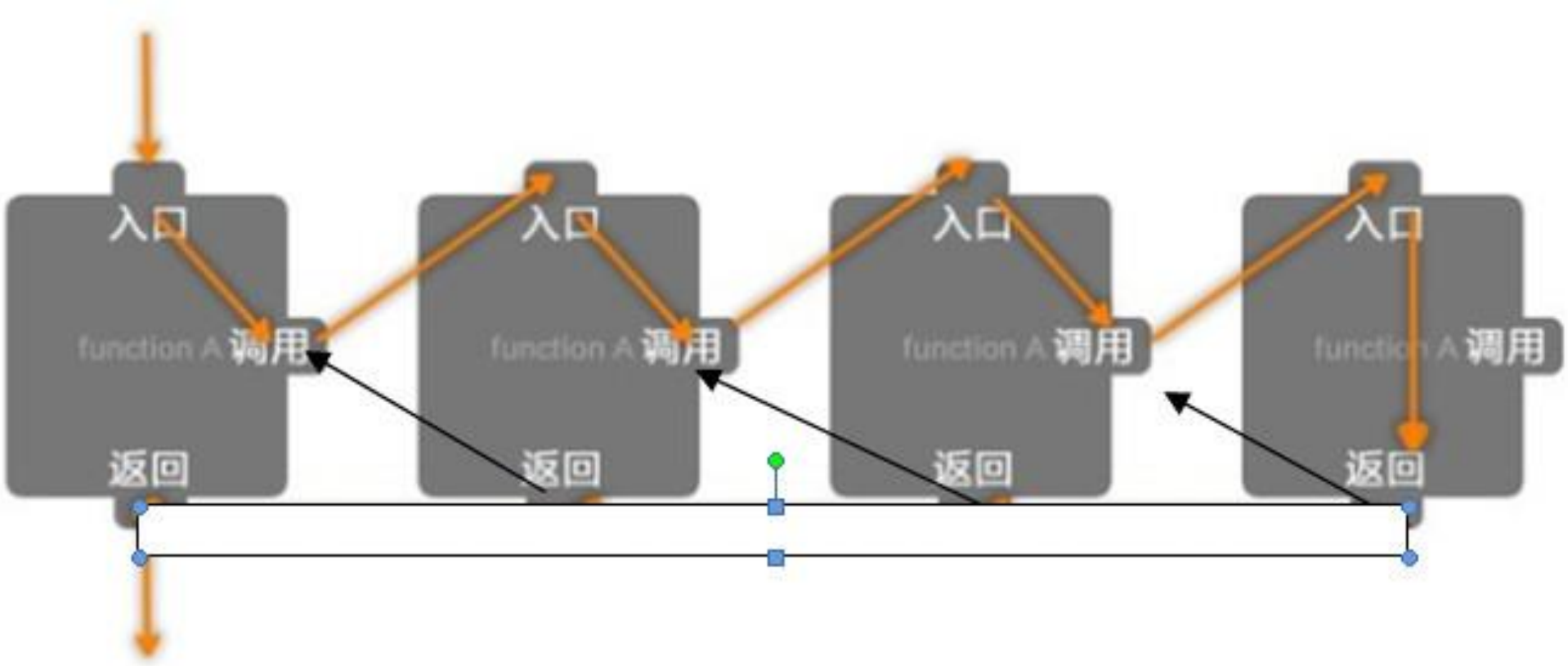
```
#include <stdio.h>
void Recursion(int depth){
    printf("抱着");
    if (!depth) printf("我的小鲤鱼");
    else Recursion(--depth);
    printf("的我");
}
int main(){
    printf("吓得我抱起了\n");
    Recursion(2);
    putchar('\n');
}
```

吓得我抱起了

抱着抱着抱着我的小鲤鱼的我的我的我

递归

递归过程中发生了什么？



递归

对比递归和递推（分别写出两种求斐波那契数列的程序）

```
int func(int x) //返回值类型声明 子程序名（参数）
{
    if(x == 0) return 1;
    If(x == 1) return 1;
    return func(x-1) + func(x-2);
}
```

```
for(int i=2;i<=n;i++)
    f[i] = f[i-1] + f[i-2];
```

分而治之

分治算法的基本思想是将一个规模为N的问题分解为K个规模较小的子问题，这些子问题相互独立且与原问题性质相同。求出子问题的解，就可得到原问题的解。即一种分目标完成程序算法，简单问题可用二分法完成。

实现手段：递归调用子程序

特点：大幅度优化程序复杂度（回忆一下二分查找的效率）

排序算法比较

空间复杂度：执行这个算法所需要的内存空间
时间复杂度：指执行算法所需要的计算工作量
计算复杂度的方法：估算——只关心复杂度的数量级

	时间复杂度	空间复杂度
选择排序	O(N2)	O(N)
冒泡排序	O(N2)	O(N)
二分查找	O(logN)	O(N)
归并排序	O(NlogN)	O(N)
快速排序	O(NlogN)	O(N)

问题背景

思考：如何快速合并两个已经有序的数列

输入：
5（n个数字）
1 3 5 7 9（ai）
2 4 6 8 10 （bi）
输出：
1 2 3 4 5 6 7 8 9 10（排好的数列）
思考：算法复杂度？
注意：不要被误导，合并的序列长度不需要相等。

核心代码

```
scanf("%d",&n);
for(int i=0;i<n;i++) scanf("%d",&a[i]);
for(int i=n;i<n*2;i++) scanf("%d",&a[i]);
int i = 0,j = n,k = 0;
while(i<n && j<n*2)
{
    if(a[i] < a[j]) r[k++] = a[i++];
    else r[k++] = a[j++];
}
while(i<n) r[k++] = a[i++];
while(j<n*2) r[k++] = a[j++];
for(i=0;i<n*2;i++) printf("%d\n",r[i]);
```

问题一般化

思考：一个无序数列的每一位数字本身是否就是有序的？

分而治之的思路：把无序数列划分为多个集合，最小的集合只有一个数字（保证有序），然后依次合并所有集合，完成整体的排序。

输入：

8（n个数字）

8 5 3 1 4 7 2 6（ai）

输出：

1 2 3 4 5 6 7 8（排好的数列）

归并演示

思考如何划分集合？

一一合并为二，二二合并为四，四四合并为八 …

待排序数据：**92 28 62 84 62 16 56 87**

第零次归并：**[92][28][62][84][62][16][56][87]**

第一次归并：**[28 92] [62 84] [16 62] [56 87]**

第二次归并：**[28 62 84 92] [16 56 62 87]**

第三次归并：**[16 28 56 62 62 84 87 92]**

反过来就是折半划分，一分为二，二分为四，四分为八

思考：最多划分为多少层？

核心代码

```
void merge(int l,int r)
{
    if(l==r) return; //长度为1直接退出
    int mid = l + r >> 1;
    merge(l,mid);
    merge(mid+1,r);
    //以下代码为二路归并
    int i = l,j = mid+1,k = l;
    while(i<=mid && j<=r)
    {
        if(a[i] < a[j]) g[k++] = a[i++];
        else g[k++] = a[j++];
    }
    while(i<=mid) g[k++] = a[i++];
    while(j<=r) g[k++] = a[i++];
    for(k=l;k<=r;k++) a[k] = g[k];
}
```

归并排序与逆序对

使用归并排序求无序数列的逆序对数量。

观察：1 3 5 / 2 4 6
逆序对数量：1 + 2 = 3
合并以后：1 2 3 4 5
逆序对数量：0

以此类推...
每个人接受速度不一样，留作课后思考。

回顾测试

- **求解逆序对**

  给定n个数字，求逆序对个数

- **变大的数字**

  给出一个十进制数字，可以将该数字相邻的两个数字进行一次交换，问交换以后可能变成的最大数字是多少？

- **和为零**

  找到n个数字中是否存在两个数字和为零

问题背景

递归的优势：快速划分！
一分为二，二分为四... 大约logN层的快速集合划分。

划分集合的标准？
归并排序没有标准，直接划分，代价为1，但是每次合并需要O(N)的二路归并代价，根据logN层划分原理，得到O(NlogN)的复杂度
快速排序的划分标准：取a[mid]为标杆，小于a[mid]的放入左边集合，大于a[mid]的放入右边集合

排序演示

待排序数据：5 1 4 3 2
第一次： 5 1 4 3 2
5>=4 且 4>=2 : 2 1 4 3 5
4>=4 且 4>=3 : 2 1 3 4 5
分为：{2 1 3} 和 {4 5}两部分
第二次： 2 1 3 和 4 5
2>=1 且 1>=1 : 1 2 3
4<=5 不需要交换
最终：1 2 3 4 5

快速排序

```
void qsort(int l,int r)
{
    int i = l,j = r;
    int k = a[(l+r) / 2];
    while(i<=j)
    {
        while(a[i] <= k) i++;
        while(a[j] >= k) j--;
        if(i<=j)
        {
            int t = a[i];
            a[i] = a[j];
            a[j] = t;
            i++;
            j--;
        }
    }
    if(i<r) qsort(i,r);
    if(l<j) qsort(l,j);
}
```

分而治之

分治算法的基本思想是将一个规模为N的问题分解为K个规模较小的子问题，这些子问题相互独立且与原问题性质相同。求出子问题的解，就可得到原问题的解。即一种分目标完成程序算法，简单问题可用二分法完成。

实现手段：递归调用子程序

特点：大幅度优化程序复杂度（回忆一下二分查找的效率）

友谊的小船

n个人，已知每个人体重。独木舟承重固定，每只独木舟最多坐两个人，可以坐一个人或者两个人。显然要求总重量不超过独木舟承重，假设每个人体重也不超过独木舟承重，问最少需要几只独木舟？

Input

第一行包含两个正整数n (0<n<=100)和m (0<m<=2000)，表示人数和独木舟的承重。

接下来n行，每行一个正整数，表示每个人的体重。体重不超过1000，并且每个人的体重不超过m。

Output

一行一个整数表示最少需要的独木舟数。

完美字符串

约翰认为字符串的完美度等于它里面所有字母的完美度之和。每个字母的完美度可以由你来分配，不同字母的完美度不同，分别对应一个1-26之间的整数。约翰不在乎字母大小写。（也就是说字母F和f）的完美度相同。给定一个字符串，输出它的最大可能的完美度。例如：dad，你可以将26分配给d，25分配给a，这样整个字符串完美度为77。

字符串是什么？ 字符的数组。

定义：char s[100];

输入：scanf(“%s”,s);

赋值：s[0] = ‘a’;

输出：printf(“%s”,s);

二分查找

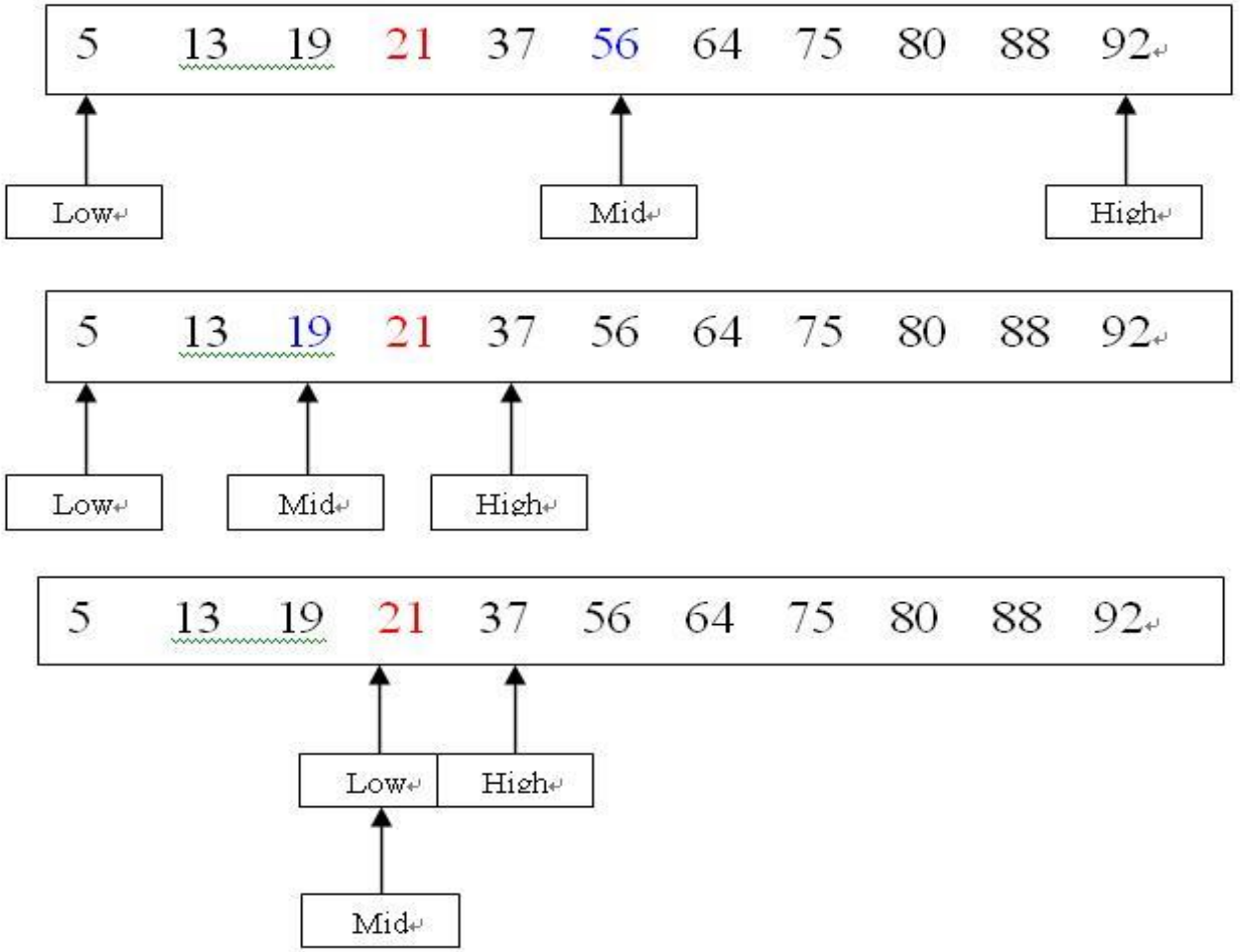
问题背景：如果在多个数字中快速确定数字x是否存在？

二分查找，也称折半查找。在有序（升序）数列中，每次拿出中间值与目标值进行比较，如果中间值和目标值相等，则找到退出；如果中间值比目标值大，则到前半段继续查找；如果中间值比目标值小，则到后半段继续查找。重复该过程直到找到与目标值相等值或者区间大小缩小为0。

特点：每次查找可以排除一半的可能性，最多查找次数？

二分查找

例子：查找21


5 13 19 21 37 56 64 75 80 88 92
Low
Mid
High
5 13 19 21 37 56 64 75 80 88 92
Low
Mid
High
5 13 19 21 37 56 64 75 80 88 92
Low
High
Mid

二分查找

```
int low = 0,high = n-1,mid,ans = -1;
while(low<=high)
{
    mid = (low + high) / 2;
    if(a[mid] == x)
    {
        ans = mid;
        break;
    }
    if(a[mid] < x)
    {
        low = mid + 1;
    }else
    {
        high = mid - 1;
    }
}
```

笨笨开车

问题：明明和聪聪准备一起去Xm远的公园玩，笨笨准备开车送他们。但是笨笨的车一次只能带一个人。已知明明和聪聪步行的速度为V1m/h，笨笨开车的速度为V2m/h(V2>V1)。那么问题来了，明明和聪聪最快什么时候可以到达公园？

Input

X V1 V2（提示：用double）

Output

ANS

习题选讲

1.hoj_A: 连续上升子序列长度
2.hoj_B: 最近的距离
3.hoj_C: 2倍增长与3倍增长
4.hoj_D: 回文素数
5.hoj_E: 水果 http://acm.hit.edu.cn/hojx/showcontest/9/E/
6.hoj_F: 直线分割平面

回顾

今天你学会了什么？

一、测试回顾
二、学习递归
三、归并排序
四、快速排序

专题训练

04-10 15:00 CST

哈尔滨工业大学程序设计实践（一）第5讲

Start Countdown: 00:02:20

Problem　Rank (120:00:00)　0 Comments　Setting　☆Favorite　Cl

Stat	#	Origin	Title
30 / 95	A	HDU 1425	sort
3 / 26	B	HDU 3743	Frosh Week
10 / 29	C	CodeForces 559B	Equivalent Strings
5 / 10	D	CodeForces 440C	One-Based Arithmetic
15 / 57	E	CodeForces 461A	Appleman and Toastman

Private. Prepared

网址：https://vjudge.net/contest/158072

密码：cxsjsj

共同进步

- 一分耕耘，一分收获！
- 希望在今后的学习生活中共同进步！

谢谢！