



# 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



# 二分

- 一、复习排序算法
- 二、复习贪心算法
- 三、超强大的二分
- 四、练习练习练习



Start Countdown: 00:16:42

Rank (1200:00:00)		Discuss	Setting	☆Favorite
#	Origin	Title		
A	HDU 1969	Pie		
B	HDU 2141	Can you find it?		
C	POJ 3273	Monthly Expense		
D	CodeForces 380A	Sereja and Prefixes		
E	CodeForces 367C	Sereja and the Arrangement of Numbers		

网址: <https://vjudge.net/contest/155109>

密码: cxsjsj



# 三种基本程序结构

## 顺序结构

顺序结构就是一条一条地从上到下执行语句，所有的语句都会被执行到，执行过的语句不会再次执行。如：赋值语句，输入输出语句。

## 选择结构

选择结构就是根据条件来判断执行哪些语句，如果给定的条件成立，就执行相应的语句，如果不成立，就执行另外一些语句。如：if条件语句。

## 循环结构

循环结构就是在达到指定条件前，重复执行某些语句。如：for循环，while循环。



# 素数判断

接受素数的审判



# 选择排序

## 接受排序的考验



# 台阶问题

一层楼有 $n$ 阶台阶，每次可以走2个台阶或者3个台阶，问一共有多少种方案？

### Input

第一行包含一个正整数 $n$  ( $2 < n \leq 100$ )。

### Output

一行一个整数表示方案个数。



# 选择排序

```
for(int i=0;i<n;i++)  
{  
    int pos;  
    for(int j=i+1;j<n;j++)  
    {  
        if(a[pos] < a[j]) pos = j;  
    }  
    swap(a[i],a[pos]);  
}
```





# 冒泡排序

算法描述：

- 1.比较相邻的元素。如果第一个比第二个大，就交换他们两个。
- 2.对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。在这一点，最后的元素应该会是最大的数。
- 3.针对所有的元素重复以上的步骤，除了最后一个。
- 4.持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。



# 冒泡排序

```
for(i = 0; i < n; i++)  
{  
    for(j = 0; j < n - i - 1; j++)  
    {  
        if(a[j] > a[j+1])  
        {  
            t = a[j];  
            a[j] = a[j+1];  
            a[j+1] = t;  
        }  
    }  
}
```



# 对比两种排序

相同点：

基于比较

不同点：

选择排序每次选择第 $i$ 小的数字，第 $i$ 轮结束后前 $i$ 个数是有序的；

冒泡排序每次筛选出第 $i$ 大的数字，第 $i$ 轮结束后后 $i$ 个数是有序的。

思考：两种排序哪种更好？（提示：复杂度？）



# 冒泡优化

```
bool flag;  
for(int i=0;i<n-1;i++)  
{  
    flag = 1;  
    for(int j=0;j<n-i-1;j++)  
    {  
        if(a[j] > a[j+1])  
        {  
            flag = 0;  
            t = a[j];  
            a[j] = a[j+1];  
            a[j+1] = t;  
        }  
    }  
    if(flag) break;  
}
```



# 冒泡排序与逆序对

逆序对：设  $A$  为一个有  $n$  个数字的有序集 ( $n > 1$ )，其中所有数字各不相同。如果存在正整数  $i, j$  使得  $1 \leq i < j \leq n$  而且  $A[i] > A[j]$ ，则  $\langle A[i], A[j] \rangle$  这个有序对称为  $A$  的一个逆序对，也称作逆序数。

问题：给定一个无序数列，如何求逆序数？

输入：

$N$

$A_i$

输出：

Ans 逆序对的数量



## 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`



包装我们的排序程序，方便后面题目的使用。

子程序

- `void xuanze(int n, int a[100]) //无返回值 子程序名（参数）`
- `{`
- 
- `}`
- `xuanze(3, a); //调用子程序`



# 贪心算法

贪心算法（又称贪婪算法）是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的是在某种意义上的局部最优解。

算法（**Algorithm**）是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。

排序是一种算法，有针对性的问题和具体的解题步骤。

贪心只是一种思想，没有确定的解题方案，甚至无法保证正确性，但是因此适用性更为广泛。





# 背包问题增强版

小李有一个可以容纳 $V$ kg的背包。现在有 $n$ 种砂石，每种总含量为 $a_i$  kg，性价比 $b_i$  bit/kg。问小李的背包最多容纳多少bit的物品？

样例输入：(按照 $V$   $N$   $a_i$   $b_i$ 的顺序输入)

100 3

60 8

30 10

80 3

样例输出：(30\*10+60\*8+10\*3=)810



# 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 $n$  ( $0 < n \leq 100$ )和 $m$  ( $0 < m \leq 2000$ )，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。



# 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？

### Input

第一行包含一个正整数 $n$  ( $0 < n \leq 10000$ )。

### Output

一行一个整数表示交换以后可能生成的最大整数。



# 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？

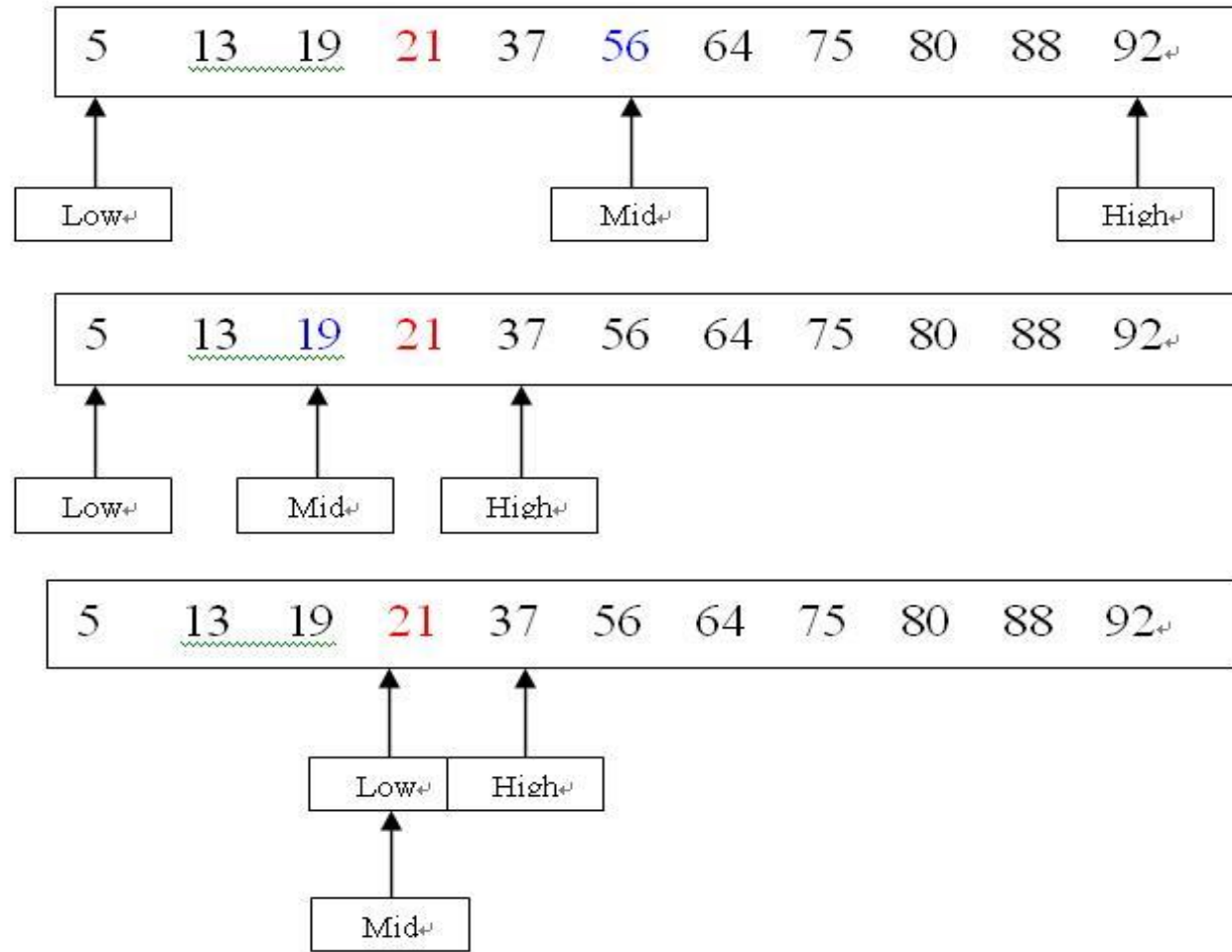
二分查找，也称折半查找。在有序（升序）数列中，每次拿出中间值与目标值进行比较，如果中间值和目标值相等，则找到退出；如果中间值比目标值大，则到前半段继续查找；如果中间值比目标值小，则到后半段继续查找。重复该过程直到找到与目标值相等值或者区间大小缩小为0。

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





# 二分查找

问题：如果在多个数字中快速确定数字 $x$ 是否存在？

输入：

$n$   $x$  ( $n$ 个数字，查找 $x$ )

$a_i$  (一共 $n$ 个数)

输出：

Yse/No

提示：可以尝试先使用遍历解决。



# 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;
    }
}
```





# 和为0

问题：给出一个长度为N的无序数组，数组中的元素为整数，有正有负包括0，并互不相等。从中找出所有和 = 0的2个数的组合，输出这2个数。

### Input

第1行，1个数N，N为数组的长度( $0 \leq N \leq 100$ )

第2 - N + 1行：A[i] ( $-10^9 \leq A[i] \leq 10^9$ )

### Output

两个数，表示答案。



## 和为0

问题：给出一个长度为N的无序数组，数组中的元素为整数，有正有负包括0，并互不相等。从中找出所有和 = 0的3个数的组合，输出这3个数。

### Input

第1行，1个数N，N为数组的长度( $0 \leq N \leq 100$ )

第2 - N + 1行：A[i] ( $-10^9 \leq A[i] \leq 10^9$ )

### Output

3个数，表示答案。



# 查找连续函数的写法

```
//x: 待查找的值, Caculate(): 所要查找的函数, 在这里单调递增
//需保证查找的值在区间范围内
double low= “区间下界”, high= “区间上界”, mid;
while(high - low > 1.0e-6)
{
    mid = (high + low)/2;
    if(Caculate(mid)<x)
        low=mid;
    else
        high=mid;
}
```



# 笨笨开车

问题：明明和聪聪准备一起去 $X$ m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 $V1$ m/h，笨笨开车的速度为 $V2$ m/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



## 在子程序中调用子程序会发生什么？

- `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`

# 作业



- 完成该PPT中问题涉及的程序。



# 今天你学会了什么？

- 一、复习排序算法
- 二、复习贪心算法
- 三、超强大的二分
- 四、练习练习练习



Start Countdown: 00:16:42

Rank (1200:00:00)		Discuss	Setting	☆Favorite
#	Origin	Title		
A	HDU 1969	Pie		
B	HDU 2141	Can you find it?		
C	POJ 3273	Monthly Expense		
D	CodeForces 380A	Sereja and Prefixes		
E	CodeForces 367C	Sereja and the Arrangement of Numbers		

网址: <https://vjudge.net/contest/155109>

密码: cxsjsj





- 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！