**实验一 算法基础**

1. **实验目的**

1．复习数据结构课程的相关知识，实现课程间的平滑过渡。

2．掌握并应用算法的数学分析方法。

3. 掌握把算法变成程序的技能。

1. **实验要求**

1.熟悉Dev C++的编程环境。

2.熟练理解这样一个观点：不同的算法能够解决相同的问题，这些算法的解题思路不同，复杂程度不同，解题效率也不同。

1. **主要仪器及耗材**

计算机及相关软件

1. **实验内容与步骤**
2. 编写程序，分别用欧几里得算法和连续整数检测算法实现求两个数的最大公约数，并回答下面的问题。
3. 欧几里得算法

**算法 Euclid(m,n)**

**//使用欧几里得算法计算gcd(m,n)**

**//输入：两个不全为0的非负整数m,n**

**//输出：m,n的最大公约数**

**while n≠0 do**

**r←m mod n**

**m←n**

**n←r**

**return m**

**问题：**

1. 对于第一个数小于第二个数的一对数字,欧几里得算法将会如何处理?该算法在处理这种输入的过程中,上述情况最多会发生几次?

第一次r等于m,m=n,n=r,相当于将m和n互换，保证m大于n。

最多发生一次。

1. a.对于所有1≤m,n≤10的输入, 即m和n都为个位数。Euclid算法最少要做几次除法?

至少需要一开始的一次除法运算。

b.对于所有1≤m,n≤10的输入, Euclid算法最多要做几次除法?

最多五次除法运算，gcd(5,8)

1. 连续整数检测算法

**算法**  **gcd(m,n)**

**//使用连续整数检测算法计算gcd(m,n)**

**//输入：两个不全为0的非负整数m,n**

**//输出：m,n的最大公约数**

**第一步：将min{m,n}的值赋给t。**

**第二步：m除以t。如果余数为0,进入第三步；否则,进入第四步。**

**第三步：n除以t。如果余数为0，返回t的值作为结果；否则，进入第四步。**

**第四步：把t的值减1。返回第二步**。

**问题：**

①.按照这个算法的当前的形式，当它的一个输入为0时，计算出来的结果是错误的，为什么？

当有一个输入为0时，t的值必然为0，数学常识，除数是不能为0的。

2. 编写程序，分别用以下两个算法实现求一个不大于给定整数n的连续质数序列。

（1）**算法 Prime(n)**

**//实现求一个不大于给定整数n的连续质数序列**

**//输入：一个整数n>1**

**//输出：包含所有小于等于n的质数的数组a**

**for i←2 to n do**

**k←sqrt(i)**

**for j←2 to k do**

**if i%j=0**

**break**

**if j>k if i>k**

**a[m]←i**

**m++**

**return a**

(2)**算法 Sieve(n)**

**//实现“埃拉托色尼筛选法”**

**//输入：一个整数n>1**

**//输出：包含所有小于等于n的质数的数组L**

**for p←2 to n do**

**A[p]←p**

**for p←2 to sqrt(n) do**

**if A[p]≠0**

**j←p\*p**

**while j<= n do**

**A[j]←0**

**j←j+p**

**//将A中剩余的元素复制到质数数组L中**

**i←0**

**for p←2 to n do**

**if A[p]≠0**

**L[i]←A[p]**

**i←i+1**

**return L**

**问题：**

1. 估计Prime算法在最坏情况下的时间复杂度?

O（n\*n）

1. 和Prime算法相比，Sieve算法是否提高了时间效率？

是的.

3.设计一个算法，在已经排序的两个列表中，找出所有相同的元素。例如，列表2,5,5和2,2,3,5,5,7，应该输出2,5,5。如果给定的两个列表的长度分别为m和n,你设计的算法的最大的比较次数是多少？编程程序，运行你设计的算法，得出正确的结果。 1 2 3 4 5

双指针遍历 最大 m+n-1次

4. 写出将十进制转换成二进制整数的标准算法。

**问题：**

①.用文字描述?

②.用伪代码描述

③.为算法写代码。

常规操作：除2取余法

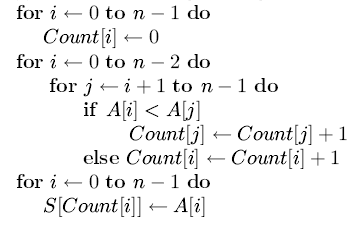
5.考虑这样一个排序算法,该算法对于待排序的数组中的每一个元素,计算比它小的元素个数,然后利用这个信息,将各个元素放到有序数组的相应位置上去.

算法：ComparisonCountingSort(A[0..n-1])

//用比较计数对数组排序

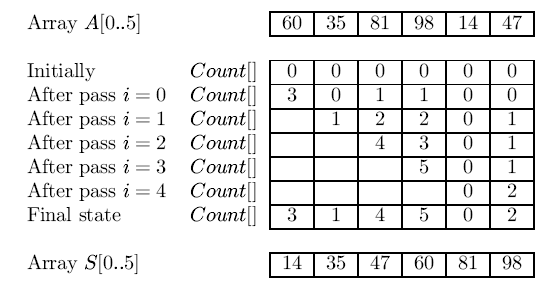
//输入：可排序数组A[0..n-1]

//输出：数组S[0..n-1]，A的元素在其中按照降序排序



**问题：**

1. 应用该算法对列表”60,35,81,98,14,47”排序



1. 该算法稳定吗?

不稳定.比如对列表”2,2\*”排序

③.该算法在位吗?（是否需要额外的存储空间）

该算法不在位.额外空间： S[] and Count[]

④.编写程序，实现以上算法，得出正确的结果。

**OK**

6..考虑下述选择排序算法：

**算法：ModSelectSort(A[0..n-1])**

**//输入：n个不等的整数的数组A[0..n-1]**

**//输出：按递增次序排序的A**

**for i←0 to n-2 do**

**for j←i+1 to n-1 do**

**if A[j]<A[i] then A[i] A[j]**

**问题：**

1. .最坏情况下该算法做多少次比较运算？

**n(n - 1)/2次**

②.最坏情况下该算法做多少次交换运算？这种情况在什么输入条件下发生？

**最坏情况，每趟排序都需要进行交换，比如完全逆序输入，需要n-1次交换。但是交换操作不能直接实现。**

**如果借助辅助变量进行交换操作，则相当于需要3\*(n-1)次交换操作。**

③. 编写程序，实现以上算法，得出正确的结果。

**OK**

7.给定正整数的数组A[1..n],测试每个元素A[i]的奇偶性.如果A[i]是奇数，则将它2倍后输出；否则直接输出A[i].

**问题：**

1. 以乘法作为基本运算，使用大O记号，还是使用大θ记号，哪个能够正确表达这个算法对于规模为n的输入所做的基本运算次数？为什么？

**由于只有奇数元素才会进行乘法操作，偶数不会。**

**那么就相当于只存在一个上界，用大O记号比较合适。**

②.如果以元素的测试作为基本运算，重复问题①?

**元素的测试所有元素都会执行到，相当于紧，用大θ记号。**

8.下述Find-Second-Min算法是找第二小算法.输入是n个不等的数构成的数组S，输出是第二小的数SecondMin.

**算法：FindSecondMin(S,n)**

**//输入：n个不等的整数的数组S[1..n]**

**//输出：第二小的数SecondMin**

**if S[1]<S[2]**

**then min←S[1];SecondMin←S[2]**

**else min←S[2];SecondMin←S[1]**

**for i←3 to n do**

**if S[i]<SecondMin**

**then if S[i]<min**

**then SecondMin←min; min←S[i]**

**else SeconMin←S[i]**

**问题：**

1. .最坏情况下该算法做多少次比较运算？

**最坏情况为：前两个元素是最大的两个，后面n-2个元素中每个元素都需要分别与第一小和第二小元素进行比较运算，需要2\*（n-2）次比较运算。**

1. .若所有输入是等概率分布的，平均情况下该算法做多少次比较？

**最好情况下：前两个元素已经是最小的，剩下n-2个元素中，每次当前元素都大于前两个元素，那么只需要n-2次比较操作。**

**平均做3/2\*（n-2）次比较运算。**

③. 编写程序，实现以上算法，得出正确的结果。

**OK**

9.考虑下面这个算法，它求的是数值数组中大小最接近的两个元素的差。

算法：MinDistance(A[0..n-1])

//输入：数字数组A[0..n-1]

//输出：数组中两个大小相差最少的元素的差值

dmin←∞

for i←0 to n-1 do

for j←0 to n-1 do

if i≠j and |A[i]-A[j]|<dmin

dmin←|A[i]-A[j]|

return dmin

**问题：**

①.这个算法的比较运算总共进行了几次？能否改进该算法？如果可以，尽可能的改进该算法。

**n个元素两两进行比较，比较运算进行了n\*(n+1)/2次，**

**明显存在重复比较的情况(一半元素），复杂度高 O(n\*n)**

**且在已知 a < b < c 而比较过了 a、b 的差的情况下，**

**没必要再比较 a 和 c 的差。**

10.请实现用贪心法解两机任务调度问题。

任务调度问题建模：

**输入**：任务集 S={1, 2, … , n}，

第 j 项任务加工时间: tj ,Z+, j=1, 2 ,…, n

**输出**：调度I，S的一个可行调度方案：1, 2, … , n 的排列 i1, i2, …, in

**目标函数**：I的完成时间，



**解I\***:使得t(I\*)达到最小，即

t(I\*)=min{t(I) | I为S的排列}。

1. **注意事项**
   * + 1. 注意掌握软件的使用方法。
       2. 注意区分大小写。
       3. 认真完成上述各题。