

# 中山大学数据科学与计算机学院本科生实验报告

## (2016 学年春季学期)

课程名称: Algorithm design

任课教师: 张子臻

年级	15	专业 (方向)	软件工程(移动信息工程)
学号	15352408	姓名	张稼伟
电话	13531810182	Email	709075442@qq.com
开始日期		完成日期	

### 目录

1.实验题目 .....	3
Soj 1051. Biker's Trip Odomete .....	3
Soj 1007 To and Fro .....	3
Soj 1036 Crypto Columns .....	3
Soj 1029 Rabbit .....	3
Soj 1028 Hanoi Tower Sequence.....	3
Soj 1206 1012 Stacking Cylinders .....	4
2.实验目的 .....	4
3.程序设计 .....	4
Soj 1051. Biker's Trip Odomete .....	4
Soj 1007 To and Fro .....	5
Soj 1036 Crypto Columns .....	5
Soj 1029 Rabbit .....	6
Soj 1028 Hanoi Tower Sequence.....	7

Soj 1206 1012 Stacking Cylinders .....	8
5.实验总结与心得 .....	9
附录、提交文件清单 .....	9

## 1.实验题目

### 模拟+其他专题

#### Soj 1051. Biker's Trip Odomete

**题意：**给车前轮直径数(inches)，转圈数(integer)和时间(seconds)，求自行车行驶距离（miles）和平均速度（in miles per hour）。

**单位换算：**

Pi= 3.1415927

5280 feet= 1 mile

12 inches = 1 foot

60 minutes = 1 hour

60 seconds = 1 minute

201.168 meters = 1 furlong

#### Soj 1007 To and Fro

**题意：**介绍了一种加密算法：将明文（要加密的信息）从左到右一列一列地写入给定列数的矩形中，再添加字符‘x’将它补满，接着从上到下一行一行蛇形(先从左到右读，然后下一行从右到左读，交替进行)地读出来形成密文（加密后的信息）。现在给你密文，要将它翻译成明文。

**约束：**列数  $2 \leq n \leq 20$ 。

#### Soj 1036 Crypto Columns

**题意：**一个字符串，去掉空格和符号后变成"plaintext"。另一个字符串叫"keyword"。将 plaintext 排成多行，每行有 keyword.size 的宽度，不能排满一列的用别的字符填满，生成了字符矩阵。每次选取 keyword 中字典序最小的字符对应的列，作为需要选取的矩阵的列，被选择过的字符不再被选择，最后形成一行新的字符串，作为输出字符串。先给出 keyword 和密文，求明文。

**约束：**keyword 长度  $n \leq 10$ ，密文长度  $m \leq 100$

#### Soj 1029 Rabbit

**题意：**开始有一对成年兔子，每对成年兔子每个月产生一对小兔子，每只小兔子经过 m 个月变成成年兔子，问经过 d 个月后有多少对兔子。

**约束：** $1 \leq m \leq 10, 1 \leq d \leq 100$

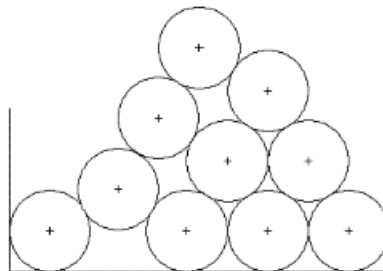
#### Soj 1028 Hanoi Tower Sequence

**题意：**汉诺塔问题，将从上到下的方向从小到大排列的盘子，从第一个

柱子中移动到另一个柱子，其中可以借助第三个柱子，并且每次移动后每根柱子从上到下方向柱子大小总是从小到大的。现在给定了一个移动规则，问第  $p$  个移动的盘子编号为多少  
 约束：  $1 \leq p \leq 10^{100}$

### Soj 1206 1012 Stacking Cylinders

**题意：**给出最底层的  $n$  个圆柱的圆心位置，圆柱半径都为 1，保证相邻两个圆柱的圆心距离在最少为 2 (so the cylinders do not overlap)，最多为 3.4(so cylinders at level  $k$  cannot touch cylinders at level  $k - 2$ )。求最顶层的圆柱的位置。一个圆柱要在上一层，它下一层必须有两个圆柱托住它。输出最顶层圆柱圆心位置包含两个浮点数  $x, y$ ， $x$  是横坐标， $y$  是高度（最底层那些圆柱圆心的高度是 1）。



约束：  $1 \leq n \leq 10$

## 2.实验目的

- 1.会做简单模拟题。
- 2.提高运用课堂所学知识解决实际问题的能力。

## 3.程序设计

### Soj 1051. Biker's Trip Odomete

**解题思路：**轮子转过的路程=直径\* $\pi$ \*圈数，平均速度=路程/时间。然后就是按照题目给的单位转换关系转换一下单位。直接列公式计算就行，没什么难度。

用到的单位转换:1hour=3600senconds, 1 mile=12\*5280inches

时间复杂度:  $O(1)$

代码:

```
#include<iostream>
#include<cstdio>
using namespace std;
#define rep(i,u,v) for (int i=(u);i<=(v);i++)
const float pi=3.1415927;
int main() {
    float diameter,time,distances,speed;
    int revolutions,T=0;
    while (cin>>diameter>>revolutions>>time,revolutions>0) {
        distances=pi*(diameter/12.0/5280.0)*revolutions;
        speed=distances/(time/60/60);
        printf("Trip #%d: %.2f %.2f\n", ++T, distances, speed);
    }
    return 0;
}
```

## Soj 1007 To and Fro

**解题思路:**这题我们只要想办法将密文恢复成矩阵，再从左到右一列一列输出来就是明文。先根据列数算出行数，然后再蛇形地一行一行填入矩阵，最后一列一列输出即可。

**时间复杂度:** $O(n)$

**代码:**

```
1 #include<iostream>
2 #include<cstring>
3 using namespace std;
4 #define rep(i,u,v) for (int i=(u);i<=(v);i++)
5 int main() {
6     int r,c,len;
7     string str;
8     while (cin>>c&&c) {
9         char a[201][201];
10        cin>>str;
11        len=str.length();
12        r=len/c;
13        int x=0,y=0; //x和y表示当前填到x行y列
14        rep(i,0,len-1) {
15            if (x%2==0) { //从左到右填
16                a[x][y]=str[i];
17                if (++y==c) {
18                    x++;y=0;
19                }
20            }
21            else { //从右到左填
22                a[x][y]=str[i];
23                if (--y==-1) {
24                    x++;y=0;
25                }
26            }
27        }
28        rep(j,0,c-1)
29            rep(i,0,r-1)
30                cout<<a[i][j];
31        cout<<endl;
32    }
33    return 0;
34 }
```

## Soj 1036 Crypto Columns

**解题思路:**简单模拟。设密文长度为 len，keyword 长度为 c，则矩阵行数为  $r=len/c$ 。我们每次找到 keyword 中字典序最小的字母，该字母在 keyword 中的位置 index 就是现在我们要填的矩阵的列，假设我们已经将密文的前 i 个字母填进去了，那么这次我们就是在第 index 列填入密文中的 i+1 到 i+c 这 c 个字母。不断重复上述操作知道填完所有字符即可。然后一行一行输出来。在这里我将这个矩阵压缩成一行，这样就可以直接使用一次字符串输出。

**时间复杂度:** $O(n)$

**代码:**

```
1 #include<cstdio>
2 #include<cstring>
3 #include<cstdlib>
4 #include<iostream>
5 using namespace std;
6 char keys[11];
7 char flag[11]; //表示这一列是否已经被处理过
8 char line[101]; //密文
9 char plaintext[101]; //明文
```

```

10 void Decrypt() { //解密
11     int i, index, column, count, keysLength, lineLength;
12     char c;
13     memset(flag, 0, sizeof(flag));
14     lineLength = strlen(line); //密文总字符数
15     keysLength = strlen(keys); //列数
16     count = lineLength / keysLength;
17     plaintext[lineLength] = 0;
18
19     //找出最小字母
20     for (column = 0; column < keysLength; column++) {
21         c = 'Z' + 1;
22         for (i = 0; i < keysLength; i++)
23             if (flag[i] == 0 && keys[i] < c)
24                 c = keys[i], index = i;
25         flag[index] = 1;
26         for (i = 0; i < count; i++) {
27             //明文的第index列
28             plaintext[keysLength * i + index] = line[count * column + i];
29         }
30     }
31 }
32 int main() {
33     while (1) {
34         gets(keys);
35         if (strcmp(keys, "THEEND") == 0) break;
36         gets(line);
37         Decrypt();
38         printf("%s\n", plaintext);
39     }
40 }

```

## Soj 1029 Rabbit

**解题思路：**设第  $i$  天有兔子  $f[i]$  对，显然  $f[i]$  由两部分组成，一部分是第  $i-1$  天已有的兔子  $f[i-1]$ ，一部分是第  $i$  天才出生的兔子。由于小兔子要经过  $m$  个月才能成年，并且一成年就可以生小兔子。所以第  $i$  天能生的兔子数就是第  $i-m$  天的兔子数  $f[i-m]$ ，所以可以得到递推公式  $f[i] = f[i-1] + f[i-m]$ 。注意，由于前  $m-1$  个月  $f[i] = f[i-1] + 1$ ，因为前  $m-1$  个月只有初始的那对兔子可以生小兔子，一开始的那对兔子对应第 0 个月，即  $f[0] = 1$ 。最后答案就是  $f[d]$ 。然后注意到当  $d=100$ ,  $m=1$  是， $f[d] = 2^{100}$ ，显然我们需要写一个高精度加法去做这题。

**时间复杂度：** $O(d*m)$ ,  $m$  为高精度的大整数长度

**代码：**

```

1 #include<iostream>
2 #include<cstring>
3 using namespace std;
4 #define rep(i,u,v) for (int i=(u);i<=(v);i++)
5 #define dto(i,u,v) for (int i=(u);i>=(v);i--)
6 struct node{
7     int a[100];
8     int len;
9     node(){}
10    node operator + (node &B){
11        int len3=len>B.len?len:B.len;
12        node n0;
13        memset(n0.a,0,sizeof(n0.a));
14        n0.len=len3;
15        rep(i,1,len3){
16            n0.a[i]=a[i]+B.a[i];
17            n0.a[i+1]+=n0.a[i]/10;
18            n0.a[i]%=10;
19        }
20        while (n0.a[n0.len+1]){
21            n0.len++;
22            n0.a[n0.len+1]+=n0.a[n0.len]/10;
23            n0.a[n0.len]%=10;
24        }
25        return n0;
26    }
27 }f[105],one;
28 int m,d;

```

```

29 int main(){
30     ios::sync_with_stdio(false);
31     f[0].a[1]=1;f[0].len=1;
32     one.a[1]=1;one.len=1;
33     while (cin>>m>>d){
34         if (!m && !d)break;
35         rep(i,1,m-1)f[i]=f[i-1]+one;
36         rep(i,m,d)f[i]=f[i-1]+f[i-m];
37         dto(i,f[d].len,1)cout<<f[d].a[i];
38         cout<<endl;
39     }
40     return 0;
41 }

```

## Soj 1028 Hanoi Tower Sequence

**解题思路:**这道题主要是求解汉诺塔第  $p$  个数的值是多少。现在我们先来看一下汉诺塔的序列关系:

1: 1  
 2: 121  
 3: 1213121  
 4: 121312141213121

细心观察可以发现,对于 1 这个数,都是隔 1 个元素出现的,对于 2 这个数,是隔 4 (也就是  $2^2$ ) 个元素,对于 3 这个数,是隔着 8 (也就是  $2^3$ ) 个元素出现的..... 可以知道对于  $n$  这个数,是隔着  $2^n$  个元素出现的。然后数字  $n$  第一次出现的位置是  $2^{(n-1)}$ 。

综上所述我们可以得出,数字  $n$  出现的位置如下:

$$2^{(n-1)} + k * (2^n) \quad k = 0, 1, 2, \dots$$

现在我们知道了位置  $p$ , 于是问题就变成了求解方程  $p = 2^{(n-1)} + k * 2^n$ 。我们的方法不断用  $p$  除以 2, 看能够整除多少次, 能够整除的次数 + 1 就是答案。这是因为:

$$[2^{(n-1)} + k * (2^n)] / (2^{(n-1)}) = 1 + 2*k$$

在这里  $p$  除以 2 除了  $(n-1)$  次, 再下一次是  $(1+2*k)$  除以 2 肯定不能整除。

**时间复杂度:**  $O(n^2)$

**代码:**

```

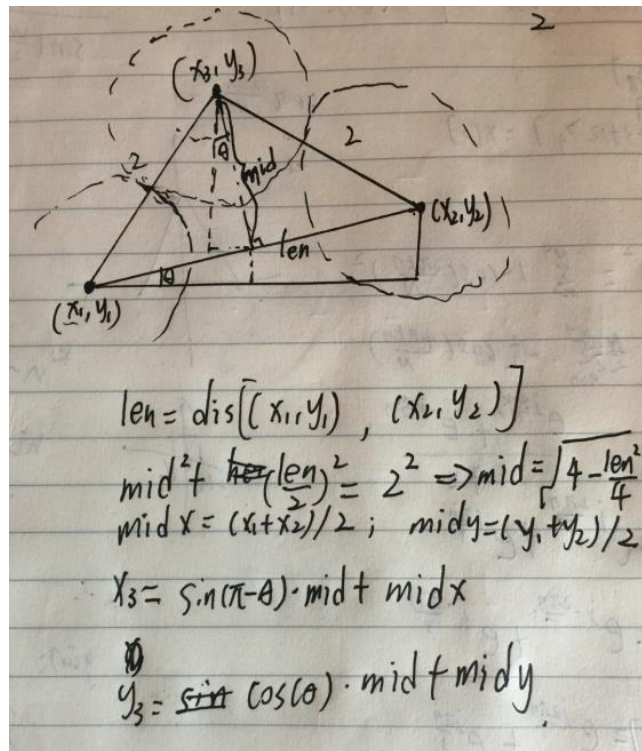
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 #define rep(i,u,v) for (int i=(u);i<=(v);i++)
5 int calc(string ss){
6     int len=(int)ss.length(),sum=0;
7     rep(i,0,len-1)ss[i] = ss[i] - '0';
8     while (ss[len-1]&2==0) { //不断/2
9         rep(i,0,len-1){
10             if (i+1<len)ss[i+1]+=(ss[i]&2)*10;
11             ss[i]=ss[i]/2;
12         }
13         sum++;
14     }
15     return sum + 1;
16 }
17
18 int main(){
19     int T,cases=0;
20     string s;
21     cin>>T;
22     while (cases!=T) {
23         cin>>s;
24         cases++;
25         cout<<"Case "<<cases<<" : "<<calc(s)<<endl;
26         if (cases!=T)cout << endl;
27     }
28     return 0;
29 }

```



## Soj 1206 1012 Stacking Cylinders

**解题思路:** 其实就是用数学几何知识, 不断通过同一层相邻两个圆柱求叠在它们上面的那个圆柱的圆心坐标。可以在草稿纸上画图得出关系式如下。



所以我们用一个循环算出每一层的圆心坐标即可。要注意本题要先对底层圆心的先按横坐标从小到大排序。

**时间复杂度:**  $O(n^2)$

**代码:**

```
1 #include<iostream>
2 #include<cstdio>
3 #include<cmath>
4 #include<algorithm>
5 using namespace std;
6 #define rep(i,u,v) for (int i=(u);i<=(v);i++)
7 struct node{
8     double x,y;
9     node(){}
10 }p[100];
11 int n;
12 bool cmp(const node &n1,const node &n2){return n1.x<n2.x;}
13 double distances(node &n1, node &n2){
14     double x=n1.x-n2.x,y=n1.y-n2.y;
15     return sqrt(x*x+y*y);
16 }
17
18 void calc(node &n1,node &n2){
19     n++;
20     double len=distances(n1,n2);
21     double mid=sqrt(4.0-len*len/4);
22     double midx=(n1.x+n2.x)/2.0;
23     double midy=(n1.y+n2.y)/2.0;
24     double sin_=(n1.y-n2.y)/len;
25     double cos_=(n2.x-n1.x)/len;
26     p[n].x=sin_*mid+midx;
27     p[n].y=cos_*mid+midy;
28 }
```



```

29 int main(){
30     int T;
31     cin>>T;
32     rep(t,1,T){
33         cin>>n;
34         if (!n)break;
35         rep(i,1,n){
36             cin>>p[i].x;
37             p[i].y=1.0;
38         }
39         sort(p+1,p+1+n,cmp);
40         int l=1,r=n;
41         while (r-l){
42             rep(i,l,r-1)calc(p[i],p[i+1]);
43             l=r+1;r=n;
44         }
45         printf("%d: %.4lf %.4lf\n",t,p[l].x,p[l].y);
46         //printf("%.4lf %.4lf\n",p[l].x,p[l].y);
47     }
48     return 0;
49 }

```

## 4.程序运行与测试

7 道题均通过成功运行并通过 sicily 的测试。

User Name	Problem	Language	Status	Run Time	Run Memory	Code Length	Submit Time
smie15352408	1012	C++	Accepted	0sec	308 KB	1262 Bytes	2017-05-29 15:39:21
smie15352408	1206	C++	Accepted	0sec	308 KB	1182 Bytes	2017-05-29 15:35:46
smie15352408	1029	C++	Accepted	0.02sec	344 KB	970 Bytes	2017-05-29 12:57:37
smie15352408	1051	C++	Accepted	0sec	308 KB	471 Bytes	2017-05-29 11:19:49
smie15352408	1028	C++	Accepted	0.04sec	308 KB	603 Bytes	2017-05-29 01:17:38
smie15352408	1036	C++	Accepted	0sec	308 KB	1042 Bytes	2017-05-29 01:10:58
smie15352408	1007	C++	Accepted	0sec	308 KB	763 Bytes	2017-05-29 01:03:41

## 5.实验总结与心得

这次题目比较简单，作为一个英语渣，我觉得更多地是考验我们能读懂题目，理解清楚题意，然后就是动手模拟实现题目要求。除了 1026 这道几何题需要用数学知识画图想清楚之外，其他都还好。专题训练结束了，希望期末能有好成绩吧。

## 附录、提交文件清单

实验报告.pdf  
 1007.cpp  
 1028.cpp  
 1029.cpp  
 1036.cpp  
 1051.cpp

1206&1012. cpp