矩阵专题 - CSDN博客

做了几个矩阵问题，总结一下。

矩阵是个很神奇的东西，有时候对于一个有规律的操作，需要执行很多次的时候，有时候可以构造矩阵很巧妙的解决。

另外对于递推式求解，可以通过构造矩阵巧妙解决。

经典的便是FIB数列，以及FIB数列的求和问题。

HDU 1575 Tr A

http://acm.hdu.edu.cn/showproblem.php?pid=1575

赤裸的矩阵快速幂乘

[cpp]view plaincopy

```
55. #include
56. #include
57. #include
58. #include
59. #include
60. #include
61. #define N 10
62. #define inf 1<<29 span>
63. #define MOD 9973
64. #define LL long long
65. usingnamespace std;
66. struct Matrix{
67. int m[N][N];
68. }init;
69. int n,k;
70. Matrix Mult(Matrix m1,Matrix m2){
71.     Matrix ans;
72. for(int i=0;i
73. for(int j=0;j
74.             ans.m[i][j]=0;
75. for(int k=0;k
76.                 ans.m[i][j]=(ans.m[i][j]+m1.m[i][k]*m2.m[k][j])%MOD;
77.         }
78. return ans;
79. }
80. Matrix Pow(Matrix m1,int b){
81.     Matrix ans;
82. for(int i=0;i
83. for(int j=0;j
84.             ans.m[i][j]=(i==j);
85. while(b){
86. if(b&1)
87.             ans=Mult(ans,m1);
88.         m1=Mult(m1,m1);
89.         b>>=1;
90.     }
91. return ans;
92. }
93. int main(){
94. int t;
95.     scanf("%d",&t);
96. while(t--){
97.         scanf("%d%d",&n,&k);
98. for(int i=0;i
99. for(int j=0;j
100.                 scanf("%d",&init.m[i][j]);
101.         init=Pow(init,k);
102. int ans=0;
103. for(int i=0;i
104.             ans=(ans+init.m[i][i])%MOD;
105.         printf("%d\n",ans);
106.     }
```

```
107. return 0;
108. }
```

HDU 1588 Gauss Fibonacci

http://acm.hdu.edu.cn/showproblem.php?pid=1588

构造矩阵，通过矩阵乘法可以得到FIB数列的某一项，矩阵为{1,1,1,0}

其中这里要求的为

F(b)+F(k+b)+F(2*k+b)……

用矩阵表示即为A^b+A^(k+b)……，可以转化为A^b*(A^(0)+A^(k)……)

将K=A^k,即A^b*(K^0+K^1+K^2……)，对于括号内部部分，有经典的二分解法，在Matrix67博客里面也有介绍

[cpp]view plaincopy

```
70. #include
71. #include
72. #include
73. #include
74. #include
75. #include
76. #define N 10
77. usingnamespace std;
78. struct Matrix{
79.     LL m[N][N];
80. }init,unit;
81. int MOD;
82. Matrix Mult(Matrix m1,Matrix m2,int n=2){
83.     Matrix ans;
84. for(int i=0;i
85. for(int j=0;j
86.             ans.m[i][j]=0;
87. for(int k=0;k
88.                 ans.m[i][j]=((LL)ans.m[i][j]+m1.m[i][k]*m2.m[k][j])%MOD;
89.         }
90. return ans;
91. }
92. Matrix Pow(Matrix m1,int b,int n=2){
93.     Matrix ans;
94. for(int i=0;i
95. for(int j=0;j
96.             ans.m[i][j]=(i==j);
97. while(b){
98. if(b&1)
99.             ans=Mult(ans,m1,n);
100.         m1=Mult(m1,m1,n);
101.         b>>=1;
102.     }
103. return ans;
104. }
105. Matrix Add(Matrix m1,Matrix m2,int n=2){
106.     Matrix ans;
107. for(int i=0;i
108. for(int j=0;j
109.             ans.m[i][j]=((LL)m1.m[i][j]+m2.m[i][j])%MOD;
110. return ans;
111. }
112. Matrix slove(Matrix init,int k,int n=2){
113. if(k==1)
114. return init;
115.     Matrix temp=slove(init,k>>1,n);
116.     temp=Add(temp,Mult(temp,Pow(init,k>>1)));
117. if(k&1)
118. return Add(temp,Pow(init,k));
119. else
120. return temp;
```

```
}
int main(){
int k,b,n;
while(scanf("%d%d%d%d",&k,&b,&n,&MOD)!=EOF){
        init.m[0][0]=init.m[0][1]=init.m[1][0]=1;
        init.m[1][1]=0;
        unit.m[0][0]=unit.m[1][1]=1;
        unit.m[1][0]=unit.m[0][1]=0;
        Matrix t;
        t=Pow(init,b);
        init=Pow(init,k);
//init.m[0][1]即第k项fib，为A^k
        init=Add(unit,slove(init,n-1)); //B=A^k,  B^0+B^1+B^2……+B^n-1
        init=Mult(init,t);
        printf("%I64d\n",init.m[0][1]);
    }
return 0;
}
```

HDU 1757 A Simple Math Problem

http://acm.hdu.edu.cn/webcontest/contest_showproblem.php?cid=1432&pid=1001&ojid=0

经典的构造矩阵解决递推式问题

[cpp]view plaincopy

```
#include
#include
#include
#include
#include
#include
#define N 10
#define inf 1<<29 span>
//#define MOD 9973
#define LL long long
#define eps 1e-7
#define zero(a) fabs(a)
#define equal(a,b) zero(a-b)
usingnamespace std;
struct Matrix{
    LL m[N][N];
}init,unit;
int MOD;
Matrix Mult(Matrix m1,Matrix m2,int n=10){
    Matrix ans;
for(int i=0;i
for(int j=0;j
            ans.m[i][j]=0;
for(int k=0;k
                ans.m[i][j]=((LL)ans.m[i][j]+m1.m[i][k]*m2.m[k][j])%MOD;
        }
return ans;
}
Matrix Pow(Matrix m1,int b,int n=10){
    Matrix ans;
for(int i=0;i
for(int j=0;j
            ans.m[i][j]=(i==j);
while(b){
if(b&1)
            ans=Mult(ans,m1,n);
        m1=Mult(m1,m1,n);
        b>>=1;
    }
return ans;
}
```

```
Matrix Add(Matrix m1,Matrix m2,int n=10){
    Matrix ans;
for(int i=0;i
for(int j=0;j
            ans.m[i][j]=((LL)m1.m[i][j]+m2.m[i][j])%MOD;
return ans;
}
Matrix slove(Matrix init,int k,int n=10){
if(k==1)
return init;
    Matrix temp=slove(init,k>>1,n);
    temp=Add(temp,Mult(temp,Pow(init,k>>1)));
if(k&1)
return Add(temp,Pow(init,k));
else
return temp;
}
void debug(Matrix m1){
for(int i=0;i<10 ispan>
for(int j=0;j<10 jspan>
            printf("%d ",m1.m[i][j]);
        printf("\n");
    }
}
int main(){
int k,b,n;
int a[10];
while(scanf("%d%d",&k,&MOD)!=EOF){
for(int i=0;i<10 ispan>
            scanf("%d",&a[i]);
if(k<10 span>
            printf("%d\n",k);
else{
            memset(init.m,0,sizeof(init.m));
for(int i=0;i<10 ispan>
                init.m[i][0]=a[i];
for(int i=1;i<10 ispan>
                init.m[i-1][i]=1;
            init=Pow(init,k-9);
            LL ans=0;
for(int i=0;i<10 ispan>
                ans=(ans+(9-i)*init.m[i][0])%MOD;
            printf("%I64d\n",ans);
        }
    }
return 0;
}
```

HDU 2157 How many ways??

http://acm.hdu.edu.cn/showproblem.php?pid=2157

又是一个经典的应用，在离散数学中有讲，可达矩阵的K次幂便是从i到j走K步能到达的方案数

[cpp]view plaincopy

```
#include
#include
#include
#include
#include
#include
#define N 25
#define inf 1<<29 span>
#define MOD 1000
```

```
92. #define LL long long
93. #define eps 1e-7
94. #define zero(a) fabs(a)
95. #define equal(a,b) zero(a-b)
96. usingnamespace std;
97. struct Matrix{
98.     LL m[N][N];
99. }init,unit;
100. Matrix Mult(Matrix m1,Matrix m2,int n=10){
101.     Matrix ans;
102. for(int i=0;i
103. for(int j=0;j
104.             ans.m[i][j]=0;
105. for(int k=0;k
106.                 ans.m[i][j]=((LL)ans.m[i][j]+m1.m[i][k]*m2.m[k][j])%MOD;
107.         }
108. return ans;
109. }
110. Matrix Pow(Matrix m1,int b,int n=10){
111.     Matrix ans;
112. for(int i=0;i
113. for(int j=0;j
114.             ans.m[i][j]=(i==j);
115. while(b){
116. if(b&1)
117.             ans=Mult(ans,m1,n);
118.         m1=Mult(m1,m1,n);
119.         b>>=1;
120.     }
121. return ans;
122. }
123. Matrix Add(Matrix m1,Matrix m2,int n=10){
124.     Matrix ans;
125. for(int i=0;i
126. for(int j=0;j
127.             ans.m[i][j]=((LL)m1.m[i][j]+m2.m[i][j])%MOD;
128. return ans;
129. }
130. Matrix slove(Matrix init,int k,int n=10){
131. if(k==1)
132. return init;
133.     Matrix temp=slove(init,k>>1,n);
134.     temp=Add(temp,Mult(temp,Pow(init,k>>1)));
135. if(k&1)
136. return Add(temp,Pow(init,k));
137. else
138. return temp;
139. }
140. void debug(Matrix m1){
141. for(int i=0;i<10 ispan>
142. for(int j=0;j<10 jspan>
143.             printf("%d ",m1.m[i][j]);
144.         printf("\n");
145.     }
146. }
147. int main(){
148. int n,m;
149. while(scanf("%d%d",&n,&m)!=EOF&&n+m){
150.         memset(init.m,0,sizeof(init.m));
151. int u,v,q,k;
152. while(m--){
153.             scanf("%d%d",&u,&v);
154.             init.m[u][v]=1;
```

```
155.        }
156.        scanf("%d",&q);
157. while(q--){
158.            scanf("%d%d%d",&u,&v,&k);
159.            Matrix unit=Pow(init,k,n);
160.            printf("%d\n",unit.m[u][v]);
161.        }
162.    }
163. return 0;
164. }
```

POJ 3233 Matrix Power Series

http://poj.org/problem?id=3233

经典矩阵二分，A^1+A^2+A^3……A^n,如果n为偶数A^1+A^2……A^n/2+A^(n/2)*(A^1+A^2……A^n/2)如果是奇数，就在最后再加一项A^n,这样就可以递归二分下去。

[cpp]view plaincopy

```
79. #include
80. #include
81. #include
82. #include
83. #include
84. #include
85. #define N 35
86. #define inf 1<<29 span>
87. //#define MOD 9973
88. #define LL long long
89. #define eps 1e-7
90. #define zero(a) fabs(a)
91. #define equal(a,b) zero(a-b)
92. usingnamespace std;
93. struct Matrix{
94. int m[N][N];
95. }init,unit;
96. int MOD;
97. Matrix Mult(Matrix m1,Matrix m2,int n){
98.     Matrix ans;
99. for(int i=0;i
100. for(int j=0;j
101.            ans.m[i][j]=0;
102. for(int k=0;k
103.                ans.m[i][j]=(ans.m[i][j]+m1.m[i][k]*m2.m[k][j])%MOD;
104.        }
105. return ans;
106. }
107. Matrix Pow(Matrix m1,int b,int n){
108.     Matrix ans;
109. for(int i=0;i
110. for(int j=0;j
111.            ans.m[i][j]=(i==j);
112. while(b){
113. if(b&1)
114.            ans=Mult(ans,m1,n);
115.        m1=Mult(m1,m1,n);
116.        b>>=1;
117.    }
118. return ans;
119. }
120. Matrix Add(Matrix m1,Matrix m2,int n){
121.     Matrix ans;
122. for(int i=0;i
123. for(int j=0;j
124.            ans.m[i][j]=(m1.m[i][j]+m2.m[i][j])%MOD;
125. return ans;
```

```
126. }
127. Matrix slove(Matrix init,int k,int n){
128. if(k==1)
129. return init;
130.     Matrix temp=slove(init,k>>1,n);
131.     temp=Add(temp,Mult(temp,Pow(init,k>>1,n),n),n);
132. if(k&1)
133. return Add(temp,Pow(init,k,n),n);
134. else
135. return temp;
136. }
137. void debug(Matrix m1,int n){
138. for(int i=0;i
139.         printf("%d",m1.m[i][0]);
140. for(int j=1;j
141.             printf(" %d",m1.m[i][j]);
142.         printf("\n");
143.     }
144. }
145. int main(){
146. int k,b,n;
147. while(scanf("%d%d%d",&n,&k,&MOD)!=EOF){
148. for(int i=0;i
149. for(int j=0;j
150.                 scanf("%d",&init.m[i][j]);
151.                 init.m[i][j]%=MOD;
152.             }
153.         debug(slove(init,k,n),n);
154.     }
155. return 0;
156. }
```

ZOJ 3497 Mistwald

同样首先判断是否可达，如果K次不可达，则必然是False。如果K次只能到达目标点，则说明是实话，如果K次有多点可达，则说明是可能

http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=3497

[cpp|view plaincopy

```
90. #include
91. #include
92. #include
93. #include
94. #include
95. #include
96. #define N 30
97. #define inf 1<<29 span>
98. //#define MOD 9973
99. #define LL long long
100. #define eps 1e-7
101. #define zero(a) fabs(a)
102. #define equal(a,b) zero(a-b)
103. usingnamespace std;
104. struct Matrix{
105. int m[N][N];
106. }init,unit;
107. Matrix Mult(Matrix m1,Matrix m2,int n){
108.     Matrix ans;
109. for(int i=0;i
110. for(int j=0;j
111.             ans.m[i][j]=0;
112. for(int k=0;k
113.                 ans.m[i][j]=(ans.m[i][j]+m1.m[i][k]*m2.m[k][j]);
114.         }
115. return ans;
```

```
}
Matrix Pow(Matrix m1,int b,int n){
    Matrix ans;
for(int i=0;i
for(int j=0;j
            ans.m[i][j]=(i==j);
while(b){
if(b&1)
            ans=Mult(ans,m1,n);
        m1=Mult(m1,m1,n);
        b>>=1;
    }
return ans;
}
Matrix Add(Matrix m1,Matrix m2,int n){
    Matrix ans;
for(int i=0;i
for(int j=0;j
            ans.m[i][j]=(m1.m[i][j]+m2.m[i][j]);
return ans;
}
int main(){
int t,c,r;
    scanf("%d",&t);
while(t--){
        scanf("%d%d",&r,&c);
        memset(init.m,0,sizeof(init.m));
for(int i=0;i
for(int j=0;j
int x1,x2,x3,x4,y1,y2,y3,y4;
                getchar();
                scanf("((%d,%d),(%d,%d),(%d,%d),(%d,%d))",&x1,&y1,&x2,&y2,&x3,&y3,&x4,&y4);
if(i==r-1&&j==c-1) continue;
                init.m[i*c+j][(x1-1)*c+y1-1]=1;
                init.m[i*c+j][(x2-1)*c+y2-1]=1;
                init.m[i*c+j][(x3-1)*c+y3-1]=1;
                init.m[i*c+j][(x4-1)*c+y4-1]=1;
            }
        }
int q,k;
//  for(int i=0;i
//      init.m[i][i]=0;
        scanf("%d",&q);
while(q--){
            scanf("%d",&k);
            Matrix tmp=Pow(init,k,r*c);
if(tmp.m[0][r*c-1]==0){
                puts("False");
continue;
            }
int cnt=0;
for(int i=0;i
if(tmp.m[0][i])
                cnt++;
if(!cnt)
                puts("True");
else
                puts("Maybe");
        }
        puts("");
    }
return 0;
}
```

HDU 2807 The Shortest Path

http://acm.hdu.edu.cn/showproblem.php?pid=2807

没啥好说的，直接搞就行了。不过挺卡时间的，注意优化

[cpp]view plaincopy

```
#include
#include
#define NN 81
#define inf 1<<29 span>
usingnamespace std;
int n,m;
int matrix[NN][NN][NN];
int dis[NN][NN];
void init(){
for(int i=1;i< nispan>
for (int j=1; j< mjspan>
for (int k=1; k< mkspan>
            scanf("%d",&matrix[i][j][k]);
}
void get_dis(){
int temp[NN][NN];
for (int a=1; a< naspan>
for (int b=1; b< nbspan>
        dis[a][b]=inf;
for (int a=1; a< naspan>
for (int b=1; b< nbspan>
if(a==b) continue;
for (int i=1; i< mispan>
for (int j=1; j< mjspan>
            temp[i][j]=0;
for (int k=1; k< mkspan>
            temp[i][j]+=matrix[a][i][k]*matrix[b][k][j];
        }
    }
for(int c=1; c< ncspan>
if(a==c||b==c)  continue;
int flag=1;
for (int i=1; i< mflagispan>
for (int j=1; j< mflagjspan>
if (temp[i][j]!=matrix[c][i][j]) flag=0;
if (flag) dis[a][c]=1;
        }
    }
}
void Floyed(){
for(int k=1;k< nkspan>
for(int i=1;i< nispan>
for(int j=1;j< njspan>
if (k==i || k==j || i==j) continue;
if (dis[i][j]>dis[i][k]+dis[k][j])
            dis[i][j]=dis[i][k]+dis[k][j];
    }
}
void Query(){
int t,x,y;
    scanf("%d",&t);
while(t--){
        scanf("%d%d",&x,&y);
if (dis[x][y]>=inf)
            printf("Sorry\n");
```

```
124. else
125.                 printf("%d\n",dis[x][y]);
126.     }
127. }
128. int main(){
129. while(scanf("%d%d",&n,&m)!=EOF&&n!=0&&m!=0){
130.         init();
131.         get_dis();
132.         Floyed();
133.         Query();
134.     }
135. return 0;
136. }
```

HDU 3483 A Very Simple Problem

http://acm.hdu.edu.cn/showproblem.php?pid=3483

贴个图，神构造，矩阵完美解决

$$(n+1)^x = \sum_{k=0}^{x} \binom{x}{k} n^x$$

设 $S(n) = \sum_{k=1}^{n} x^k k^x$，构造向量

$$T(n) = [x^n n^0, x^n n^1, \dots, x^n n^{x-1}, x^n n^x, S(n)]$$

那么 $T(n+1) = [x^{n+1}(n+1)^0, x^{n+1}(n+1)^1, \dots, x^{n+1}(n+1)^{x-1}, x^{n+1}(n+1)^x, S(n+1)]$

对比一下就可以构造出矩阵 A

$$\begin{pmatrix} x\binom{0}{0} & x\binom{1}{0} & x\binom{2}{0} & \cdots\cdots & x\binom{x-1}{0} & x\binom{x}{0} & x\binom{x}{0} \\ 0 & x\binom{1}{1} & x\binom{2}{1} & \cdots\cdots & x\binom{x-1}{1} & x\binom{x}{1} & x\binom{x}{1} \\ 0 & 0 & x\binom{2}{2} & \cdots\cdots & x\binom{x-1}{2} & x\binom{x}{2} & x\binom{x}{2} \\ & & & \square\square\square & & & \\ & & & \square\square\square & & & \\ & & & \square\square\square & & & \\ 0 & 0 & 0 & \cdots\cdots & 0 & x\binom{x}{x} & x\binom{x}{x} \\ 0 & 0 & 0 & \cdots\cdots & 0 & 0 & 1 \end{pmatrix}$$

[cpp]view plaincopy

```
85. #include
86. #include
87. #include
88. #include
89. #include
90. #include
91. #define N 55
92. #define inf 1<<29 span>
93. //#define MOD 9973
94. #define LL long long
95. #define eps 1e-7
96. #define zero(a) fabs(a)
97. #define equal(a,b) zero(a-b)
98. usingnamespace std;
99. struct Matrix{
100.     LL m[N][N];
101. }init,unit;
102. LL MOD;
103. Matrix Mult(Matrix m1,Matrix m2,int n){
104.     Matrix ans;
105.     memset(ans.m,0,sizeof(ans.m));
106. for(int k=0;k
107. for(int i=0;i
108. if(m1.m[i][k])
109. for(int j=0;j
```

```
                ans.m[i][j]+=m1.m[i][k]*m2.m[k][j];
if(ans.m[i][j]>=MOD)ans.m[i][j]%=MOD;
            }
/*for(int i=0;i
    for(int j=0;j
        ans.m[i][j]=0;
        for(int k=0;k
            ans.m[i][j]=(ans.m[i][j]+m1.m[i][k]*m2.m[k][j])%MOD;
    }*/
return ans;
}
Matrix Pow(Matrix m1,int b,int n){
    Matrix ans;
for(int i=0;i
for(int j=0;j
        ans.m[i][j]=(i==j);
while(b){
if(b&1)
        ans=Mult(ans,m1,n);
    m1=Mult(m1,m1,n);
    b>>=1;
    }
return ans;
}
Matrix Add(Matrix m1,Matrix m2,int n){
    Matrix ans;
for(int i=0;i
for(int j=0;j
        ans.m[i][j]=(m1.m[i][j]+m2.m[i][j])%MOD;
return ans;
}
int n,x;
LL c[N][N];
int main(){
while(scanf("%d%d%d",&n,&x,&MOD)!=EOF){
if(n==-1&&x==-1&&MOD==-1) break;
    memset(init.m,0,sizeof(init.m));
for(int i=0;i< xispan>
        c[i][0]=1;c[i][i]=1;
for(int j=1;j
            c[i][j]=(c[i-1][j]+c[i-1][j-1]);
if(c[i][j]>=MOD)
                c[i][j]-=MOD;
        }
    }
for(int j=0;j< xjspan>
for(int i=0;i< jispan>
            init.m[i][j]=(c[j][i]*x)%MOD;
for(int i=0;i< xispan>
        init.m[i][x+1]=(x*c[x][i])%MOD;
    init.m[x+1][x+1]=1;
    init=Pow(init,n-1,x+2);
    LL ans=0;
for(int i=0;i< xispan>
        ans=(ans+(LL)x*init.m[i][x+1])%MOD;
    printf("%I64d\n",ans);
    }
return 0;
}
```

HDU 2276 Kiki & Little Kiki 2

每一个位置的状态ai=(ai+ai-1)%2，可以用异或加速

构造矩阵，便可解决

1 1 0 0 0 0 0

0 1 1 0 0 0 0

0 0 1 1 0 0 0

0 0 0 1 1 0 0

0 0 0 0 1 1 0

0 0 0 0 0 1 1

1 0 0 0 0 0 1

http://acm.hdu.edu.cn/showproblem.php?pid=2276

[cpp]view plaincopy

```
70. #include
71. #include
72. #include
73. #include
74. #include
75. #include
76. #define N 105
77. #define inf 1<<29 span>
78. #define MOD 9973
79. #define Max 301
80. #define LL long long
81. #define eps 1e-7
82. #define zero(a) fabs(a)
83. #define equal(a,b) zero(a-b)
84. usingnamespace std;
85. struct Matrix{
86. int m[N][N];
87. }init,unit;
88. int n,k;
89. Matrix Mult(Matrix m1,Matrix m2){
90.     Matrix ans;
91. for(int i=0;i
92. for(int j=0;j
93.             ans.m[i][j]=0;
94. for(int k=0;k
95.                 ans.m[i][j]=(ans.m[i][j]+(m1.m[i][k]*m2.m[k][j]))%2;
96.         }
97. return ans;
98. }
99. Matrix Pow(Matrix m1,int b){
100.     Matrix ans;
101. for(int i=0;i
102. for(int j=0;j
103.             ans.m[i][j]=(i==j);
104. while(b){
105. if(b&1)
106.             ans=Mult(ans,m1);
107.         m1=Mult(m1,m1);
108.         b>>=1;
109.     }
110. return ans;
111. }
112. void debug(Matrix m1){
113. for(int i=0;i
114. for(int j=0;j
115.             printf("%d ",m1.m[i][j]);
116.         printf("\n");
117.     }
118. }
119. char str[N];
120. int main(){
121. while(scanf("%d",&k)!=EOF){
122.         scanf("%s",str);
123.         n=strlen(str);
```

```
124.        memset(init.m,0,sizeof(init.m));
125. for(int i=0;i
126.            init.m[i][i]=init.m[(i-1+n)%n][i]=1;
127.        init=Pow(init,k);
128. //debug(init);
129. for(int i=0;i
130. int t=0;
131. for(int j=0;j
132.                t=t^((str[j]-'0')*init.m[j][i]);
133.            printf("%d",t);
134.        }
135.        puts("");
136.    }
137. return 0;
138. }
```

HDU 2855 Fibonacci Check-up

http://acm.hdu.edu.cn/showproblem.php?pid=2855

这个完全就是神构造，完全想不到，或者可以通过打表，打规律得到

$$\sum_{k=0}^{n} C_n^k F(k) = \frac{1}{\sqrt{5}}[\sum_{k=1}^{n}(\frac{1+\sqrt{5}}{2})^k C_n^k - \sum_{k=1}^{n}(\frac{1-\sqrt{5}}{2})^k C_n^k]$$

$$= \frac{1}{\sqrt{5}}[(\frac{3+\sqrt{5}}{2})^n - (\frac{3-\sqrt{5}}{2})^n]$$

$$= \frac{1}{\sqrt{5}}[(\frac{6+2\sqrt{5}}{4})^n - (\frac{6-2\sqrt{5}}{4})^n]$$

$$= \frac{1}{\sqrt{5}}\{[\frac{(1+\sqrt{5})^2}{4}]^n - [\frac{(1-\sqrt{5})^2}{4}]^n\}$$

$$= \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^{2n} - (\frac{1-\sqrt{5}}{2})^{2n}]$$

$$= F(2n)$$

有了这个结论，熟悉矩阵的同学，肯定就想到用矩阵快速乘法去解决。

因为根据 fibonacci 的通项公式，有：

$$[F(n), F(n-1)] = [F(n-1), F(n-2)]\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

对以上式子加以迭代，便有 $$[F(n), F(n-1)] = [F(1), F(0)]\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{n-1}$$

注意对 n=0 的特殊讨论即可。复杂度 O(8*log(10^9)),1s 内足够出解了

[cpp|view plaincopy

```
54. #include
55. #include
56. #include
57. #include
58. #include
59. #include
60. #define N 2
61. usingnamespace std;
62. struct Matrix{
63. int m[N][N];
64. }init,unit;
65. int n=2;
66. int MOD;
67. Matrix Mult(Matrix m1,Matrix m2){
68.     Matrix ans;
69.     memset(ans.m,0,sizeof(ans.m));
70. for(int i=0;i
71. for(int k=0;k
72. if(m1.m[i][k])
73. for(int j=0;j
```

```
74.                     ans.m[i][j]=(ans.m[i][j]+(m1.m[i][k]*m2.m[k][j]))%MOD;
75.         }
76. return ans;
77. }
78. Matrix Pow(Matrix m1,int b){
79.     Matrix ans;
80. for(int i=0;i
81. for(int j=0;j
82.             ans.m[i][j]=(i==j);
83. while(b){
84. if(b&1)
85.             ans=Mult(ans,m1);
86.         m1=Mult(m1,m1);
87.         b>>=1;
88.     }
89. return ans;
90. }
91. int main(){
92. int t,k;
93.     scanf("%d",&t);
94. while(t--){
95.         scanf("%d%d",&k,&MOD);
96. if(k==0){
97.             printf("0\n");
98. continue;
99.         }
100.         init.m[0][0]=init.m[0][1]=init.m[1][0]=1;
101.         init.m[1][1]=0;
102.         init=Pow(init,2*k-1);
103.         printf("%d\n",init.m[0][0]);
104.     }
105. return 0;
106. }
```