**数论**

**GCD:**

int gcd(int a,int b)

{

int t;

for(;b;t=b,b=a%b,a=t);

return a;

}

**exGCD:**

在已知a, b求解一组x，y使得a\*x+b\*y=Gcd(a,b)

int exgcd(int a,int b,int &x,int &y)

{

int res;

if(b==0)

{

x=1;

y=0;

res=a;

}

else

{

res=exgcd(b,a%b,x,y);

int t=x;

x=y;

y=t-a/b\*y;

}

return res;

}

**伪素数判断（64位）：**

long long multiply( long long a,long long b,long long n)

{

long long exp,res=0;

//这个就是用加法，来模拟乘法，为了防止运算的数太大，

exp=a%n;

while(b)

{

if(b&1)

{

res+=exp;

if(res>n)

res-=n;

}

exp<<=1;

if(exp>n)

exp-=n;

b/=2;

}

return res;

}

long long paw(long long a,long long t,long long n)

{

long long ans=1;

a=a%n;

while(t)

{

if(t%2==1)

ans=multiply(ans,a,n);

a=multiply(a,a,n);

t/=2;

}

return ans;

}

bool miller\_rabin(long long n,long long ti)

{

if(n==2)

return true;

if(n<2||!(n&1))

return false;

long long p=n-1;

long long k=0,a,x;

while(p%2==0)

{

p/=2;

k++;

}

long long ans;

srand(time(0));

for(int t=1; t<=ti; t++)

{

a=rand()%(n-1)+1;

x=paw(a,p,n);

for(int i=1; i<=k; i++)

{

ans=multiply(x,x,n);

if(ans==1&&x!=1&&x!=n-1)

return false;

x=ans;

}

if(ans!=1)

//注意这个判断！

return false;

}

return true;

}

**分解质因数（pollar\_rho）：**

（需要用到伪素数判断）

分解出的数存在pri里面

调用：search(n,180);

long long top=0,pri[100];

long long pollard\_rho(long long a,long long k,long long n)

{

srand(time(0));

long long x=rand()%(n-1)+1;

long long i=1,t=2;

long long y=x;

while(1)

{

i++;

x=(multiply(x,x,n)+k)%n;

long long ans=gcd(y-x,n);

if(ans>1&&ans<n)

return ans;

if(x==y)

return n;

if(t==i)

{

y=x;

t\*=2;

}

}

}

void search(long long n,long long k)

{

if(miller\_rabin(n,10))

{

pri[++top]=n;

return ;

}

else

{

long long p=n;

while(p>=n)

p=pollard\_rho(p,k--,n);

search(p,k);

search(n/p,k);

}

}

**快速组合数：**

double lnchoose(int n, int m)

    {

     if (m > n)

         {

              return 0;

          }

         if (m < n/2.0)

          {

              m = n-m;

         }

         double s1 = 0;

         for (int i=m+1; i<=n; i++)

         {

             s1 += log((double)i);

         }

         double s2 = 0;

         int ub = n-m;

         for (int i=2; i<=ub; i++)

         {

             s2 += log((double)i);

         }

         return s1-s2;

    }

    double choose(int n, int m)

    {

         if (m > n)

         {

             return 0;

        }

         return exp(lnchoose(n, m));

    }

**高斯消元法：**

int N;

double mat[maxn][maxn];

double x[maxn];

bool GaussCMax()

{

double d,t;

double mx;

int mid;

for(int i=0;i<N-1;i++)

{

if(mat[i][i]==0) return false;

mx=mat[i][i];mid=i;

for(int j=i+1;j<N;j++)

{

if(mat[j][i]>mx)

{

mx=mat[j][i];

mid=j;

}

}

if(mx==0.0)

return false;

if(mid!=i) //exchange row mid<->row i

{

for(int j=i;j<N+1;j++)

{

t=mat[i][j];

mat[i][j]=mat[mid][j];

mat[mid][j]=t;

}

}

for(int j=i+1;j<N+1;j++)

{

d=mat[j][i]/mat[i][i];

for(int k=i;k<N+1;k++)

mat[j][k]-=mat[i][k]\*d;

}

}

double tmp;

for(int i=N-1;i>=0;i--)

{

tmp=mat[i][N];

for(int j=i+1;j<N;j++)

tmp-=mat[i][j]\*x[j];

x[i]=tmp/mat[i][i];

}

return true;

}