努力努力再努力x

大数因式分解 Pollard_rho 算法详解

给你一个大数n，将它分解它的质因子的乘积的形式。

首先需要了解Miller_rabin判断一个数是否是素数

大数分解最简单的思想也是试除法，这里就不再展示代码了，就是从2到sqrt(n)，一个一个的试验，直到除到1或者循环完，最后判断一下是否已经除到1了即可。

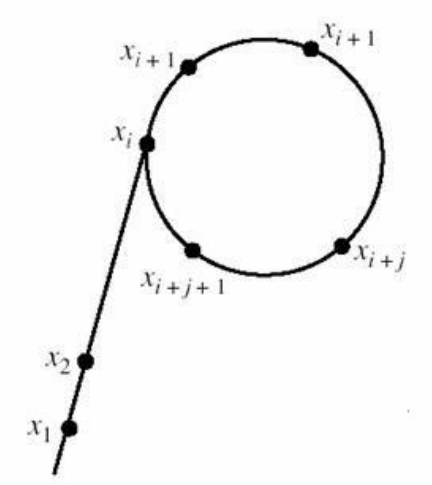
但是这样的做的复杂度是相当高的。一种很妙的思路是找到一个因子（不一定是质因子），然后再一路分解下去。这就是基于Miller_rabin的大数分解法Pollard_rho大数分解。

Pollard_rho算法的大致流程是 先判断当前数是否是素数（Miller_rabin）了，如果是则直接返回。如果不是素数的话，试图找到当前数的一个因子（可以不是质因子）。然后递归对该因子和约去这个因子的另一个因子进行分解。

那么自然的疑问就是，怎么找到当前数n的一个因子？当然不是一个一个慢慢试验，而是一种神奇的想法。其实这个找因子的过程我理解的不是非常透彻，感觉还是有一点儿试的意味，但不是盲目的枚举，而是一种随机化算法。我们假设要找的因子为p，他是随机取一个x1，由x1构造x2，使得｛p可以整除x1-x2 && x1-x2不能整除n｝则p=gcd（x1-x2，n），结果可能是1也可能不是1。如果不是1就找寻成功了一个因子，返回因子；如果是1就寻找失败，那

么我们就要不断调整x2，具体的办法通常是x2=x2*x2+c（c是自己定的）直到出现x2出现了循环==x1了表示x1选取失败重新选取x1重复上述过程。（似乎还存在一个每次找寻范围*2的优化，但是不太懂。。。）

因为x1和x2再调整时最终一定会出现循环，形成一个类似希腊字母rho的形状，故因此得名。



另外通过find函数来分解素数，如果找到了一个素数因子则加入到因子map中，否则如果用Pollard找到一个因子则递归去找素数因子。

```
#include<iostream>
#include<ctime>
#include<algorithm>
#include<map>
using namespace std;
typedef long long ll;
map<ll, int>m;
const int mod = 10000019;
const int times = 50;//测试50次
ll mul(ll a, ll b, ll m)
//求a*b%m
{
    ll ans = 0;
    a %= m;
    while(b)
    {
        if(b & 1)ans = (ans + a) % m;
        b /= 2;
        a = (a + a) % m;
    }
    return ans;
}
ll pow(ll a, ll b, ll m)
//a^b % m
```

BZOJ(50)
Codeforces(3)
Contest_Self(5)
SGU刷题之路(8)
URAL刷题之路(2)
动态规划(43)
动态规划---背包DP(22)
动态规划---概率DP(3)
动态规划---基础DP(13)
动态规划---区间DP(3)
动态规划---树形DP(2)
动态规划---状态压缩DP(3)
更多

随笔档案
2019年4月(1)
2018年10月(5)
2018年9月(52)
2018年8月(6)
2018年7月(16)
2018年6月(2)
2018年5月(93)
2018年4月(239)

```
{
    ll ans = 1;
    a %= m;
    while(b)
    {
        if(b & 1)ans = mul(a, ans, m);
        b /= 2;
        a = mul(a, a, m);
    }
    ans %= m;
    return ans;
}
bool Miller_Rabin(ll n, int repeat)//n是测试的大数，repeat是测试重复次数
{
    if(n == 2 || n == 3)return true;//特判
    if(n % 2 == 0 || n == 1)return false;//偶数和1

    //将n-1分解成2^s*d
    ll d = n - 1;
    int s = 0;
    while(!(d & 1)) ++s, d >>= 1;
    //srand((unsigned)time(NULL));在最开始调用即可
    for(int i = 0; i < repeat; i++)//重复repeat次
    {
        ll a = rand() % (n - 3) + 2;//取一个随机数,[2,n-1)
        ll x = pow(a, d, n);
        ll y = 0;
        for(int j = 0; j < s; j++)
        {
            y = mul(x, x, n);
            if(y == 1 && x != 1 && x != (n - 1))return false;
            x = y;
        }
        if(y != 1)return false;//费马小定理
    }
    return true;
}
ll gcd(ll a, ll b)
{
    return b == 0 ? a : gcd(b, a % b);
}
ll pollard_rho(ll n, ll c)//找到n的一个因子
{
    ll x = rand() % (n - 2) + 1;
    ll y = x, i = 1, k = 2;
    while(1)
    {
        i++;
        x = (mul(x, x, n) + c) + n;//不断调整x2
        ll d = gcd(y - x, n);
        if(1 < d && d < n)
            return d;//找到因子
        if(y == x)
            return n;//找到循环，返回n，重新来
        if(i == k)//一个优化
        {
            y = x;
            k <<= 1;
        }
    }
}
void Find(ll n, ll c)
{
```

2018年3月(25)

最新评论

1. Re:POJ-1700 Crossing River---过河问题（贪心）

请问如何证明这个贪心策略的正确性呢?

--vcjmhg

2. Re:KM算法（运用篇）

一次增广路中求出的slack值会更准确，循环次数比全局变量更少 为啥啊?

--T_Orang

3. Re:组合数取模方法总结（Lucas定理介绍）

%%%%%%%%%%%%

--Rain罗

4. Re:最小生成树之kruskal算法

%%%,这个启发式合并必须赞

--咯咯的C

5. Re:组合数取模方法总结（Lucas定理介绍）

orz
txdy！！！

--huyinghao

阅读排行榜

1. 数论专题（一）数论基本概念(18414)

2. 最小生成树之prim算法(16300)

3. 大数因式分解 Pollard_rho 算法详解(13460)

4. 单源最短路径---Dijkstra算法(11144)

```
88      if(n == 1)return;//递归出口
89
90      if(Miller_Rabin(n, times))//如果是素数，就加入
91      {
92          m[n]++;
93          return;
94      }
95
96      ll p = n;
97      while(p >= n)
98          p = pollard_rho(p, c--);//不断找因子，知道找到为止，返回n说明没找到
99
100     Find(p, c);
101     Find(n / p, c);
102 }
103 int main()
104 {
105     ll n;srand((unsigned)time(NULL));
106     while(cin >> n)
107     {
108         m.clear();
109         Find(n, rand() % (n - 1) + 1);//这是自己设置的一个数
110         cout<<n<<" = ";
111         for(map<ll ,int>::iterator it = m.begin(); it != m.end();)
112         {
113             cout<<it->first<<" ^ "<<it->second;
114             if((++it) != m.end())
115                 cout<<" * ";
116         }
117         cout<<endl;
118     }
119     return 0;
120 }
121 }
```

NOIP普及组、提高组培训，有意可加微信fu19521308684

分类： ACM基础篇， 数学， 数学---数论

_努力努力再努力x
关注 - 2
粉丝 - 63
+加关注

3 0

« 上一篇： 大素数测试的Miller-Rabin算法

» 下一篇： hdu-2879 hehe---积性函数

posted @ 2018-05-16 19:28 _努力努力再努力x 阅读(13461) 评论(1) 编辑 收藏

5. KM算法（运用篇）(10378)

评论排行榜

1. 组合数取模方法总结（Lucas定理介绍）(7)
2. SGU刷题之路开启(2)
3. KM算法（运用篇）(2)
4. 最小生成树之prim算法(2)
5. 大数因式分解 Pollard_rho 算法详解(1)

推荐排行榜

1. 组合数取模方法总结（Lucas定理介绍）(5)
2. KM算法（运用篇）(5)
3. 大数因式分解 Pollard_rho 算法详解(3)
4. 网络流（二）最大流的增广路算法(3)
5. EOJ-3300 奇数统计（高维前缀和）(2)