



# 努力努力再努力x

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

## 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 $x_1$ ，由 $x_1$ 构造 $x_2$ ，使得  $\{p \text{ 可以整除 } x_1-x_2 \ \&\& \ x_1-x_2 \text{ 不能整除 } n\}$  则 $p=\gcd(x_1-x_2, n)$ ，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

### 公告

Flag Counter

昵称：\_努力努力再努力x  
园龄：3年1个月  
粉丝：63  
关注：2  
[+加关注](#)

< 2021年5月 >						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

### 搜索

[找找看](#)

[谷歌搜索](#)

### 随笔分类

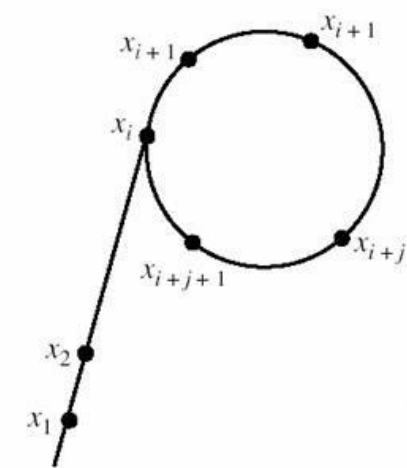
[ACM基础篇\(330\)](#)

[ACM进阶篇\(16\)](#)

[ACM准备篇\(4\)](#)

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。

```
1 #include<iostream>
2 #include<ctime>
3 #include<algorithm>
4 #include<map>
5 using namespace std;
6 typedef long long ll;
7 map<ll, int>m;
8 const int mod = 10000019;
9 const int times = 50; //测试50次
10 ll mul(ll a, ll b, ll m)
11 //求a*b%m
12 {
13     ll ans = 0;
14     a %= m;
15     while(b)
16     {
17         if(b & 1)ans = (ans + a) % m;
18         b /= 2;
19         a = (a + a) % m;
20     }
21     return ans;
22 }
23 ll pow(ll a, ll b, ll m)
24 //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)

```
25 {
26     ll ans = 1;
27     a %= m;
28     while(b)
29     {
30         if(b & 1)ans = mul(a, ans, m);
31         b /= 2;
32         a = mul(a, a, m);
33     }
34     ans %= m;
35     return ans;
36 }
37 bool Miller_Rabin(ll n, int repeat)//n是测试的大数, repeat是测试重复次数
38 {
39     if(n == 2 || n == 3)return true;//特判
40     if(n % 2 == 0 || n == 1)return false;//偶数和1
41
42     //将n-1分解成2^s*d
43     ll d = n - 1;
44     int s = 0;
45     while(!(d & 1)) ++s, d >>= 1;
46     //srand((unsigned)time(NULL));在最开始调用即可
47     for(int i = 0; i < repeat; i++)//重复repeat次
48     {
49         ll a = rand() % (n - 3) + 2;//取一个随机数, [2,n-1]
50         ll x = pow(a, d, n);
51         ll y = 0;
52         for(int j = 0; j < s; j++)
53         {
54             y = mul(x, x, n);
55             if(y == 1 && x != 1 && x != (n - 1))return false;
56             x = y;
57         }
58         if(y != 1)return false;//费马小定理
59     }
60     return true;
61 }
62 ll gcd(ll a, ll b)
63 {
64     return b == 0 ? a : gcd(b, a % b);
65 }
66 ll pollard_rho(ll n, ll c)//找到n的一个因子
67 {
68     ll x = rand() % (n - 2) + 1;
69     ll y = x, i = 1, k = 2;
70     while(1)
71     {
72         i++;
73         x = (mul(x, x, n) + c) % n;//不断调整x2
74         ll d = gcd(y - x, n);
75         if(1 < d && d < n)
76             return d;//找到因子
77         if(y == x)
78             return n;//找到循环, 返回n, 重新来
79         if(i == k)//一个优化
80         {
81             y = x;
82             k <<= 1;
83         }
84     }
85 }
86 void Find(ll n, ll c)
87 {
```

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

好文要顶

关注我

收藏该文

\_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)

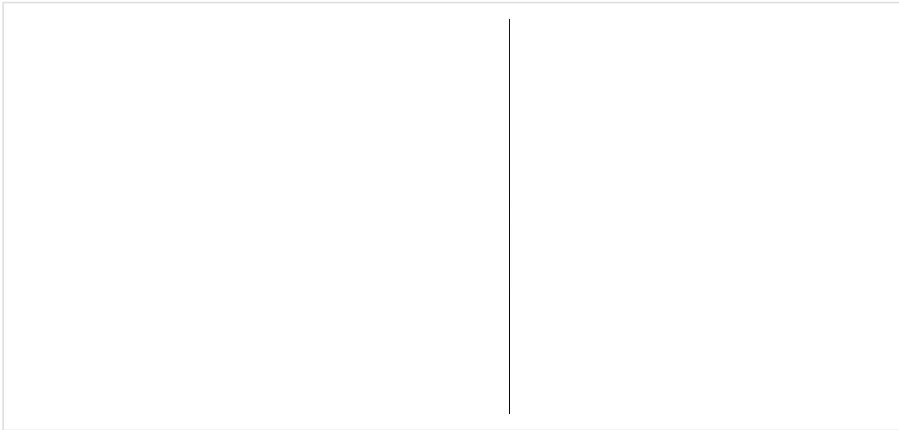
推荐排行榜

- 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)

刷新评论 刷新页面 返回顶部

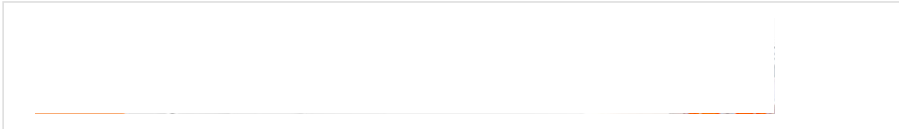
登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

- 【推荐】阿里云爆品销量榜单出炉，精选爆款产品低至0.55折
- 【推荐】7大类400多种组件，HarmonyOS鸿蒙三方库来了，赶紧收藏！
- 【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载！
- 【推荐】限时秒杀！国云大数据魔镜，企业级云分析平台



园子动态：

- 致园友们的一封检讨书：都是我们的错
- 数据库实例 CPU 100% 引发全站故障
- 发起一个开源项目：博客引擎 fluss



最新新闻：

- 菜鸟：2021财年全年收入372.5亿元，同比增68%
  - 阿里第四财季营收1874亿元，净亏损54.79亿元
  - 小鹏汽车：Q1营收29.5亿元 净亏损7.866亿元
  - B站发布Q1财报：总营收39亿元，同比增长68%
  - HTC发布VIVE FOCUS 3等系列新品 虚拟代言人也来了
- » 更多新闻...

