阶

**定义**



|  |  |  |  |
| --- | --- | --- | --- |
| 对于一个正整数 ，求满足  质。这个最小正整数 称作 在对值十分明确的时候，可以记做  **性质** | 的最小正整数  取模意义下的阶，记做  。 | 。保证  ，在 | 互  的 |
| 1. 充要条件是   1. 若  ，则 2. 若 ，则 3. 令  ，则 | 。推论：  。  对 取模两两不同。 | 。 |  |
| **原根** |  |  |  |
| **定义** |  |  |  |
| 若 是模 意义下的原根，则 | 满足 。 |  |  |

**性质**



1. 模 意义下存在原根，当且仅当 是如下形式的数： 。（ 为奇素数, 为正整数
2. 当 为奇素数时，模 意义下的原根个数为 
3. 若p是一个奇素数， 是模 的一个原根，则 和 是模  的原根；若 是模 的一个原根，则 是模 的原根



1. 对于质数 ， ，将 分解质因数，得到 



，则正整数 是模 意义下的原根的充分必要条件是：对于所有 ，



  。证明：充分性很显然。必要性：首先考虑阶的第 1 点性

质，可以得知  ，那么，如果这个值比 小，必然可以找



到一个 ，使得 ，那么 ，故 不是原根，否则，说明 ， 是原根。



const int N = 1000005; int cnt, tot, p;

int vis[N], prime[N], fac[N];

//质数筛

void Factor(int x) { tot = 0;

int t = (int) sqrt(x + 0.5);

for(int i = 1; prime[i] <= t; i++) { if(x % prime[i] == 0) {

fac[tot++] = prime[i];

while(x % prime[i] == 0) x /= prime[i];

}

}

if(x > 1) fac[tot++] = x;

}

int mypow(int a, int b, int mod) { int ans = 1;

while(b) {

if(b & 1) ans = 1ll\*ans\*a% mod; a = 1ll\*a \* a % mod;

b>>= 1;

}

return ans;

}

init();

int solve(int \_p) { p=\_p;

Factor(p - 1);

for(int g = 2; g < p; g++) { bool flag = true;

for(int i = 0; i < tot; i++) { int t = (p - 1) / fac[i]; if(quick\_pow(g, t, p) == 1) {

flag = false; break;

}

}

if(flag) {return g;break;}

}

}