

解方程

一. 考察内容:

哈希

二. 题目分析:

[题目大意]

解出方程 $a_0 + a_1 * x^1 + a_2 * x^2 + \dots + a_n * x^n = 0$ 在 $[1, m]$ 的解。

[写题思路]

首先, 考虑枚举 x 的解, 需要枚举 $1-m$ 的所有情况, 期望得分30。

考虑优化枚举的方式, 我们可以找一些大素数, 求出预处理出在这些素数的模意义下的解。这时再枚举每种情况时, 对于一个 x , 只需要判断是否每个 $x \% \text{prime}$ 都满足在模 prime 意义下的有解即可, 时间复杂度 $O(\text{模数大小} * \text{模数个数} * n + m)$, 常数为找的素数个数, 期望得分100。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 08 07
*文件类型: 源代码文件
*题目来源: COGS
*当前状态: 已通过
*备忘录: 哈希 数论
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
using namespace std;
namespace IO
{
#define SIZE 1
int T, end;
char s[SIZE+1];
int in()
{
    T=0;
    return end=fread(s, 1, SIZE, stdin);
}
int read(int &a)
{
    int i=a=0, flag=1;
    while(1)
    {
        if(T==end&&!in()) return i;
        if(s[T]==' ' || s[T]=='\n' || s[T]=='\r')
        {
            if(i)return i;
        }
        else if(s[T]=='-') flag=-1;
        else
    }
}
```

```

        {
            a=flag*(flag*a*10+s[T]-'0');
            ++i;
        }
        ++T;
    }
}
int read(char a[])
{
    int i=0;
    while(1)
    {
        if(T==end&&!in()) return i;
        if(s[T]==' '||s[T]=='\n' ||s[T]=='\r')
        {
            if(i)return i;
        }
        else a[i++]=s[T];
        ++T;
    }
}
}
#define MAXN 110
#define MAXM 1000010
//99991,99989,99971,99961,99929
int MOD,n,m,p[10]={10007,10093,10957,12917,13187},vis[100001]
[10],ans[MAXN],_ans,mod[MAXN][10];
int add(int a,int b)
{
    return (a+b)%MOD;
}
int mul(int a,int b)
{
    return (long long)a*b%MOD;
}
struct BigInt
{
    char a[10005];
    int len,flag;
    void read()
    {
        len=I0::read(a+1);
        if(a[1]=='-')
        {
            flag=1;
            for(int i=1;i<len;++i)a[i]=a[i+1];
            --len;
        }
        for(int i=1;i<=len;++i)a[i]-='0';
    }
    int mod()
    {
        int ans=0;
        for(int i=1;i<=len;++i)ans=add(mul(ans,10),a[i]);
        return flag?-ans:ans;
    }
}a[MAXN];
/* Variable explain:
*/
void read()
{
    using namespace I0;
    freopen("equationa.in","r",stdin);
    freopen("equationa.out","w",stdout);
    // freopen("in","r",stdin);

```

```

    // freopen("out","w",stdout);
    read(n);
    read(m);
    for(int i=0;i<=n;++i)
    {
        a[i].read();
    }
    return;
}
bool judge(int x,int m)
{
    int ans=0,pow=1;
    for(int i=0;i<=n;++i)
    {
        ans=add(ans,mul(mod[i][m],pow));
        pow=mul(pow,x);
    }
    return ans==0;
}
void init()
{
    for(int j=0;j<5;++j)
    {
        MOD=p[j];
        for(int i=0;i<=n;++i)
            mod[i][j]=a[i].mod();
    }
    for(int i=0;i<5;++i)
    {
        MOD=p[i];
        if(a[0].a[i]==0)vis[MOD][i]=1;
        for(int j=1;j<min(m,MOD);++j)
        {
            vis[j][i]=judge(j,i);
        }
    }
}
int main()
{
    read();
    init();
    MOD=p[0];
    int _bas=0,bas[100000],T=1;
    for(int i=1;i<=m;++i)
    {
        if(vis[i%p[0]][0]&&vis[i%p[1]][1]&&vis[i%p[2]][2]&&vis[i%p[3]][3]&&vis[i%p[4]]
[4])
        {
            ans[++_ans]=i;
        }
    }
    printf("%d\n",_ans);
    for(int i=1;i<=_ans;++i)printf("%d\n",ans[i]);
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)