怎样跑得更快 解题报告

湖北省武汉市第二中学 吕凯风

1 试题来源

我和陶润洲同学一起出的题。可以在这里找到：http://uoj.ac/problem/62。

2 试题大意

大力水手问禅师："大师，我觉得我光有力气是不够的。比如我吃菠菜可以让力气更大，但是却没有提升跑步的速度。请问怎样才能跑得更快？我试过吃白菜，没有效果。"

禅师浅笑，答："方法很简单，不过若想我教你，你先看看这道 UOJ Round 的 C 题。"

令 $p = 998244353$（$7 \times 17 \times 2^{23} + 1$，一个质数）。

给你整数 n, c, d。现在有整数 $x_1, \ldots, x_n$ 和 $b_1, \ldots, b_n$，满足 $0 \le x_1, \ldots, x_n < p$，$0 \le b_1, \ldots, b_n < p$，且对于 $1 \le i \le n$ 满足：

$$\sum_{j=1}^{n} \gcd(i, j)^c \cdot \mathrm{lcm}(i, j)^d \cdot x_j \equiv b_i \pmod p \tag{1}$$

其中 $v \equiv u \pmod p$ 表示 v 和 u 除以 p 的余数相等。$\gcd(i, j)$ 表示 i 和 j 的最大公约数，$\mathrm{lcm}(i, j)$ 表示 i 和 j 的最小公倍数。

有 q 个询问，每次给出 $b_1, \ldots, b_n$，请你解出 $x_1, \ldots, x_n$ 的值。如果有多组解输出任意一组即可。如果无解输出 -1。

对于所有数据，$nq \le 3 \times 10^5$，$0 \le c, d \le 10^9$。

<table>
<tr><th>编号</th><th>n</th><th>q</th><th>其他</th></tr>
<tr><td>1</td><td>≤ 3</td><td>$= 1$</td><td>$c, d \le 1000$</td></tr>
<tr><td>2</td><td rowspan="7">≤ 100</td><td>$= 1$</td><td>$c = d$</td></tr>
<tr><td>3</td><td rowspan="2">≤ 10</td><td rowspan="2">保证有唯一解</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td rowspan="4">≤ 1000</td><td>$c = 1, d = 0$</td></tr>
<tr><td>6</td><td rowspan="3">保证有唯一解</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td rowspan="6">≤ 1000</td><td rowspan="3">$= 1$</td><td rowspan="3">保证有唯一解</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td><td rowspan="3">≤ 10</td><td>$c = d$</td></tr>
<tr><td>13</td><td rowspan="2">保证有唯一解</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td><td rowspan="6">$\le 10^5$</td><td rowspan="6">≤ 3</td><td rowspan="2">$c = 1, d = 0$</td></tr>
<tr><td>16</td></tr>
<tr><td>17</td><td>$c = d$</td></tr>
<tr><td>18</td><td rowspan="3">无</td></tr>
<tr><td>19</td></tr>
<tr><td>20</td></tr>
</table>

时间限制：1s

空间限制：256MB

后记：还没听完题，大力水手就嘶吼着：“太难了我不会我不会！”，飞快地跑掉了。禅师看着大力水手消失的背影，叹了口气说：“你们这些人啊，每天就想做些大水题，一碰到难题，跑得不知道比谁都快。”后来大力水手把 UOJ Round 的 C 题题面贴在了汽车的后挡风玻璃上，人类从此掌握了光速旅行的正确方式。

3 算法介绍

4 算法一

对于第 1 个测试点，$n \le 3$。手解方程就行了！

可以通过 1 号测试点获得 5 分。

5 算法二

对于第 2, 12, 17 号测试点，$c = d$，由于 $\gcd(i, j)\,\mathrm{lcm}\,(i, j) = ij$，所以：

$$\sum_{j=1}^{n} i^c j^c \cdot x_j \equiv b_i \pmod p \tag{2}$$

得到：

$$\sum_{j=1}^{n} j^c \cdot x_j \equiv b_i \cdot i^{-c} \pmod p \tag{3}$$

所以右边的 $b_i \cdot i^{-c}$ 都必须相等，不相等就无解，相等就 $x_1 \equiv b_i \cdot i^{-c} \pmod p$，其它令为 0 就行了。

可以通过 2, 12, 17 号测试点获得 15 分。

6 算法三

显然我们可以高斯消元。直接上高斯消元是 $O(qn^3)$ 的。

可以通过前 4 个测试点获得 20 分。

7 算法四

显然我们可以用高斯消元预处理逆矩阵（不满秩的情况要稍微处理下）。然后就能做到 $O(n^3 + qn^2)$ 了。

可以通过前 8 个测试点获得 40 分。

进行一些常数优化，可以通过前 14 个测试点获得 70 分。

8 算法五

首先，前面提到过，学过小学奥数的我们知道：$\mathrm{lcm}(i,j)=\frac{ij}{\gcd(i.j)}$。所以这题其实是：

$$\sum_{j=1}^{n}\gcd(i,j)^{c-d}\cdot i^{d}\cdot j^{d}\cdot x_j=b_i \tag{4}$$

但是其实这种题都可做：

$$\sum_{j=1}^{n}f(\gcd(i,j))\cdot g(i)\cdot h(j)\cdot x_j=b_i \tag{5}$$

可能很多人的注意力都在“这玩意儿怎么解啊”，其实只要换个姿势问问自己“要是有人告诉了我 x，我应该怎么验证它是对的呢？”这题就可做了。

其实关键的坑人的地方在于 $f(\gcd(i,j))$。假设我有一个函数 $f_r(n)$，满足 $f(n)=\sum_{d|n}f_r(d)$，其中 $d\mid n$ 表示 d 是 n 的约数。这个式子的意思是，$f(n)$ 等于所有 n 的约数带进 f_r 后的和。知道 f 后 f_r 是很好搞的，因为 $f_r(n)=f(n)-\sum_{d|n\text{且}d\neq n}f_r(d)$，所以就能递推了。妈呀，怎么枚举约数啊？其实只要这样搞就行了：

```
for (int i = 1; i <= n; i++)
    f_r[i] = f[i];
for (int i = 1; i <= n; i++)
    for (int j = i + i; j <= n; j += i)
        f_r[j] -= f_r[i];
```

看起来是 $O(n^2)$ 的？好吧如果你不知道这是 $O(n\log n)$ 的话就是个悲伤的故事。由于 $\sum_{i=1}^{n}\frac{1}{i}=O(\log n)$，所以 $\sum_{i=1}^{n}\frac{n}{i}=O(n\log n)$。

为什么要这样？因为我们知道如果 $d\mid\gcd(i,j)$ 那么肯定有 $d\mid i$ 且 $d\mid j$，反之亦然。这样就把讨厌的 gcd 给去掉了。

所以我们可以写出这样的等式：

$$\sum_{j=1}^{n}\sum_{d}[d\mid i][d\mid j]\cdot f_r(d)\cdot g(i)\cdot h(j)\cdot x_j=b_i \tag{6}$$

这里有个符号：$[P]$ 表示表达式 P 为真时值为 1 否则为 0。比如 $[d\mid i]$ 的值在 $d\mid i$ 时为 1 否则为 0。

接下来怎么办？好像并没有简化问题。我们把两个求和符号反过来，得到：

$$\sum_{d}\sum_{j=1}^{n}[d \mid i][d \mid j] \cdot f_r(d) \cdot g(i) \cdot h(j) \cdot x_j = b_i \tag{7}$$

然后我们移一下项：

$$\sum_{d \mid i} f_r(d) \sum_{j=1}^{n}[d \mid j] \cdot h(j) \cdot x_j = b_i/g(i) \tag{8}$$

仔细观察，发现 $\sum_{j=1}^{n}[d \mid j] \cdot h(j) \cdot x_j$ 的意思是把所有 j 是 d 的倍数的 $h(j) \cdot x_j$ 加起来。反正这个只跟 d 的值有关，我们记为 z_d。于是我们得到：

$$\sum_{d \mid i} f_r(d) z_d = b_i/g(i) \tag{9}$$

这个式子的意思是，对于每个 i，把所有 d 是 i 的约数的 $f_r(d)z_d$ 加起来，得到结果 $b_i/g(i)$。现在我们知道右边，想求左边。似曾相识，对不对？

```
for (int i = 1; i <= n; i++)
    f_z[i] = b[i] / g(i);
for (int i = 1; i <= n; i++)
    for (int j = i + i; j <= n; j += i)
        f_z[j] -= f_z[i];
```

这样得到的 $f_z[d]$ 就是 $f_r(d)z_d$。想得到 z_d？由于 $f_r(d)$ 已经求出，所以 $z_d = f_z[d]/f_r(d)$。

但是 z_d 并不是最终答案。回忆 z_d 的表达式：

$$z_d = \sum_{j=1}^{n}[d \mid j] \cdot h(j) \cdot x_j \tag{10}$$

现在知道左边，想求右边，怎么办？是不是还是似曾相识呢！由于我们知道 $h(d) \cdot x_d = z_d - \sum_{j=1}^{n}[j > d][d \mid j] \cdot h(j) \cdot x_j$，所以：

```
for (int i = 1; i <= n; i++)
    hx[i] = z[d];
for (int i = n; i >= 1; i--)
    for (int j = i + i; j <= n; j += i)
        hx[i] -= hx[j];
```

嗯，现在我们知道了 $h(j)x_j$，那么 x_j 就好求了。

由于中间过程涉及了除法，所以就会带来无解和多解的情况。对于本题 $g(i)$ 和 $h(j)$ 肯定都不是 p 的倍数，所以都有逆元。而 $f_r(d)$ 可能没有逆元。这种情况假如 $f_z[d] \neq 0$ 那么显然无解，如果 $f_z[d] = 0$ 就有多组解，我们把 z_d 随便附个值比如让 $z_d = 0$ 就好了。

罗嗦了半天其实跟算法五本质是一样的。这题其实就是把 b 除以 $g(i)$ 然后莫比乌斯反演，然后除以 f 的莫比乌斯反演，再进行莫比乌斯反演，再除以 $h(j)$，三个莫比乌斯反演掷地有声。

9 数据生成方式

大部分数据直接随机生成即可，由于 $f_r(n) = n \cdot \prod_p (1 - \frac{1}{p^{\alpha(c-d)}})$，所以就可以对着构造几个函数值模 p 为 0 的数据，就可以产生无解和多解的情况了。

10 参考程序

- `equation.cpp` 是参考程序。