

DAY 07 题解

---褚战 贺一鸣

#A、又是约数之和

题目：

#A、又是约数之和

题目描述

给定正整数 N

记 S_i 为正整数 i 的约数之和

令

$$T_i = (-1)^i \times i$$

你需要求

$$\sum_{i=1}^N (S_i - T_i + A)$$

的结果

输入格式

输入三个整数 N, A

输出格式

输出一个整数表示 $\sum_{i=1}^N (S_i - T_i + A)$

输入样例

```
10 1
```

输出样例

```
91
```

数据规模

对于 10% 的数据 $1 \leq N \leq 1000$

对于 20% 的数据 $1 \leq N \leq 10^5$

对于 70% 的数据 $1 \leq N \leq 8 \times 10^6$

对于 100% 的数据 $1 \leq N, A \leq 2 \times 10^7$

这是一道数论题，正解是用欧拉筛，第一重循环 i ，如果 i 为质数，则 $s[i]$ 为 $i+1$ ，接下来第二重循环 j ，如果 $i \% \text{prime}[j] \neq 0$ ， $s[i * \text{prime}[j]] = s[i] * s[\text{prime}[j]]$ ，否则我们发现好像 $s[i * \text{prime}[j]]$ 并不好求，于是我们在一个数组 num ， i 是质数时 $\text{num}[i] = i+1$ ， $i \% \text{prime}[j] \neq 0$ 时， $\text{num}[i * \text{prime}[j]] = s[\text{prime}[j]]$ ， $i \% \text{prime}[j] = 0$ 时， $s[i * \text{prime}[j]] = s[i] / \text{num}[\text{prime}[j]] * (\text{num}[\text{prime}[j]] * \text{prime}[j] + 1)$ ， $\text{num}[i * \text{prime}[j]] = \text{num}[\text{prime}[j]] * \text{prime}[j] + 1$ ，接下来的代码按题目描述的写就行了，时间复杂度 $O(n)$ 。

#C、解密

题目：

#C、解密

题目描述

给定一个正整数 k , 有 k 次询问

每次给定三个正整数 n_i, e_i, d_i , 求两个正整数 p_i, q_i , 使 $n_i = p_i \times q_i$ 、 $e_i \times d_i = (p_i - 1)(q_i - 1) + 1$

输入格式

第一行一个正整数 k , 表示有 k 次询问

接下来 k 行, 第 i 行三个正整数 n_i, d_i, e_i

输出格式

输出 k 行, 每行两个正整数 p_i, q_i 表示答案

为使输出统一, 你应当保证 $p_i \leq q_i$

如果无解, 请输出 `NO`

输入样例

```
10
770 77 5
633 1 211
545 1 499
683 3 227
858 3 257
723 37 13
572 26 11
867 17 17
829 3 263
528 4 109
```

输出样例

```
2 385
NO
NO
NO
11 78
3 241
2 286
NO
NO
6 88
```

数据规模

以下记 $m = n - e \times d + 2$

保证对于 100% 的数据, $1 \leq k \leq 10^5$, 对于任意的 $1 \leq i \leq k$, $1 \leq n_i \leq 10^{18}$, $1 \leq e_i \times d_i \leq 10^{18}$, $1 \leq m \leq 10^9$

测试点 1 ~ 2, $k \leq 10^3$, $n \leq 10^3$, $m \leq 10^3$

测试点 3 ~ 4, $k \leq 10^3$, $n \leq 10^9$, $m \leq 6 \times 10^4$

测试点 5 ~ 6, $k \leq 10^3$, $n \leq 10^9$, $m \leq 10^9$

测试点 7 ~ 10, $k \leq 10^5$, $n \leq 10^{18}$, $m \leq 10^9$

这题在数据规模那里，第一句话其实是给的提示，我们开始推式子：它这里要求 $e \cdot d = (p-1) \cdot (q-1) + 1$ ，把后面括号拆掉， $e \cdot d = p \cdot q - p - q + 2$ ，然后我们还知道 $n = p \cdot q$ ，接着我们再把那个提示带进去可以得出 $m = p + q$ ，题目要求 p, q 都要是正整数，也就是要大于等于 1，也就是说 m 小于 2 肯定无解，再根据 $m = p + q$ 这个式子我们可以都成个方，接着我们把 $(p+q)^2$ 和 $(q-p)^2$ 作对比，发现差 4 个 n ，我们又得到了一个式子， $m^2 - 4n = (p-q)^2$ ，然后等式两边各加一个根号，这样我们就得出了 p, q 的和和差， p, q 也求出来了，在判断 p, q 是否满足条件，满足输出 p, q ，否则输出 NO，就解决了。

#B、V形点

题目描述

平面有 n 个互不相同的点,第 i 个点记为 p_i

其中 p_i 的坐标为 (x_i, y_i)

V 形点定义为:

- 由 3 个点 p_x, p_y, p_z 组成
- 其中 p_x 到 p_y 的距离与 p_y 到 p_z 的距离相等

(x_1, y_1) 与 (x_2, y_2) 距离为

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

请你统计能构成多少个 V 形点

输入格式

第一行输入一个整数 n

第二行输入 n 个整数 x_i

第三行输入 n 个整数 y_i

输出格式

输出一个整数,表示答案

输入样例

```
3
0 1 2
0 0 0
```

输出样例

```
2
```

数据规模

对于 20% 的数据, $1 \leq n \leq 100, 0 \leq x_i, y_i \leq 100$

对于 40% 的数据, $1 \leq n \leq 800, -500 \leq x_i, y_i \leq 500$

对于 100% 的数据, $1 \leq n \leq 2 \times 10^3, -10^4 \leq x_i, y_i \leq -10^4$

保证所有点互不相同

这道题 n 只到 2000，我们可以先开一个结构体数组存每一个点的坐标，然后每次选择一个点 i ，开一个 `unordered_map` 作为桶存任意一点到 i 的距离，然后遍历一遍 `map` 累加到 `cnt` 变量里就可以了。

```
for(int i = 1;i <= n;i++)
{
    unordered_map<int,int>m;
    for(int j = 1;j <= n;j++)
    {
        m[a[j].dis(a[i])]++;
    }
    for(auto &&j : m)
    {
        cnt += j.second * (j.second - 1);
    }
}
```

#D、跑团机器人

题目描述

在桌面角色扮演游戏(*TRPG* ,俗称“跑团”)中,玩家需要掷出若干个骰子,根据掷出的结果推进游戏进度

在线上同样可以跑团,方法是由玩家们向机器人发出指令,由机器人随机产生每个需要掷出的骰子的结果

玩家向机器人发出的指令是一个仅涉及加法和减法的表达式,即对若干个数字进行一系列加法或减法计算。这些数字可以是直接给出的非负整数(数字不超过 1000),也可以是若干个骰子掷出的结果

“掷骰子”这个动作对应的指令格式为 xdy ,表示摇动 x 个 y 面的骰子($1 \leq x \leq 1000, 2 \leq y \leq 1000$)

当 x 为 1 时, 1 可以省略

例如指令 `2d3+3-d4` 的意思是: 先掷出 2 个 3 面骰子(你不必考虑现实中是否存在这样的骰子),不妨假设结果为 1 和 3,则 `2d3` 的结果就是两个骰子的面值之和 4

然后计算 $4 + 3$,得到结果为 7

再掷出 1 个 4 面骰子,不妨假设结果为 2,则计算 $7 - 2$ 得到最终结果 5

本题就请你计算玩家输入的指令里,不同种类的骰子需要掷出几个,以及可能得到的结果在什么区间范围内

输入格式

输入在一行中给出一条符合题目描述的玩家输入机器人的指令。题目保证指令长度不超过 2×10^4

输出格式

首先输出不同种类的骰子分别需要掷出几个

每种骰子的信息占一行,依次输出骰子的面数和投掷的数量,按面数从小到大输出

输入指令保证至少有一个骰子需要掷出

最后一行输出两个数,表示根据输入指令可以得到的最小结果和最大结果

同一行数字间以 1 个空格分隔,行首尾不得有多余空格

样例输入

```
d6+3d5+2-2d3+2d5
```

样例输出

```
3 2
5 5
6 1
2 31
```

这题其实是一道模拟题，需要对于给出的字符串进行处理。我们要先维护符号变量 a 初始化为 1（正 1 负 -1），4 个变量 x, y, maxn, minn 用于存储数字和最大最小值，同时开一个 map 存储骰子的数量，初始化为 0。从前往后遍历字符串，遇到数字存在 x 里，遇到 + 或 - 更改 a 并更改 maxn 和 minn，遇到字符 d 往后读 y，然后更改 maxn 和 minn，最后遍历 map 输出（按面数升序输出），输出最小值和最大值。

关键部分：

```
if(s[l] == 'd')
{
    if(x == 0)
        x = 1;
    l++;
    while(l <= r && s[l] != '+' && s[l] !=
'-')
        y = y * 10 + s[l++] - '0';
    cnt[y] += x;
    if(d == 1)
    {
        maxn += x * y;
```

```
        minn += x;
    }
    else
    {
        maxn -= x;
        minn -= x * y;
    }
}
```

(l 为当前遍历位置, r 为结束位置)