

NOIP 提高组模拟题 0v0 专场

题目名称	数论(0v0)	摘 Galo(OvO)	吃草(0 π 0)
输入文件	STD	STD	STD
输出文件	STD	STD	STD
测试点个数	10	10	10
测试点分值	10	10	10
内存限制	512M	512M	512M
时间限制	1S	1S	2S



1.数论(0v0)

1.1 题目描述:

聪明的 0v0 正在学习莫比乌斯反演。



她看到了这样的一道题：有 $n*m$ 个人站成了一个 $n*m$ 的方阵……

剩下的题面，聪明的 0v0 不记得了。但是，她通过自己高超的数论技巧，给出了一个转化后的模型：给出 n 和 m ，求

$$\sum_{i=1}^n \sum_{j=1}^m \min(\lfloor \frac{n}{i} \rfloor, \lfloor \frac{m}{j} \rfloor) * [gcd(i, j) = 1]$$

聪明的 0v0 当然知道怎么做了，但是她想考考你。

1.2 输入格式:

一行三个正整数 n , m , p 。

1.3 输出格式

一行一个非负整数，设答案为 x ，输出 $x \bmod p$ 。

1.4 样例输入

1 2 998244353

1.5 样例输出

2

1.6 样例解释

聪明的 0v0 不需要样例解释。

1.7 数据范围

30% $n, m \leq 2000$ $p = 998244353$ 。

30% $n * m \leq 10^9$ $n, m \leq 10^5$ p 为质数

20% $n, m \leq 10^6$ p 为质数

20% $n, m \leq 10^7$ p 为合数

对于所有数据，保证 $p \leq 10^9$

2.摘 Galo(OvO)

2.1 题目描述:

OvO 在野外看到了一棵 Galo 树，看到食物的 OvO 瞪大了眼睛，变成了 OvO。



这棵 Galo 树可以看做是一棵以 1 号点为根的 n 个点的有根数，除了根节点以外，每个节点 i 都有一个 Galo，美味度为 $w[i]$ 。

OvO 发现，如果她摘下了 i 号 Galo，那么 i 的子树中的 Galo 以及 i 到根的路径上的其他 Galo 都会死掉。

OvO 的袋子只能装 k 个 Galo，她的嘴巴里还能叼 1 个，请问她所摘 Galo 的美味度之和的最大值是多少？

2.2 输入格式:

第一行两个正整数 n, k 。

第二行到第 n 行，第 i 行两个正整数 $f[i], w[i]$ ，表示 i 号点的父亲为 $f[i]$ （保证 $x[i] < i$ ），第 i 个 Galo 美味度为 $w[i]$ 。

2.3 输出格式

一行一个非负整数，为最大美味值。

2.4 样例输入

```
4 1
1 10
2 3
2 6
```

2.5 样例输出

10

2.6 数据范围

30% $n, k \leq 200$

30% $n * k * k \leq 10^7$

40% $n * k \leq 10^7$

对于所有数据, $n, k, w[i] \leq 10^5$

2.7 样例解释

尽管 **OvO** 最多可以摘两个 **Galo**, 但是最优情况是只摘下第二个点的 **Galo**, 美味度为 10。

3.吃草(0 π 0)

3.1 题目描述:

New Orleans 家的后院有很多片草坪，Sullivan 负责清理过高的草。但是，Sullivan 还有很多家务要干，于是，她想到了一个好方法。



后院总共有 n 片草坪，第 i 片草坪投影到数轴上，是一段 $l[i]$ 到 $r[i]$ 的闭区间，保证 $l[i]+r[i]$ 是偶数， $l[i] \leq r[i]$ 。

Sullivan 可以在整点上放 0v0 来把草吃掉（于是 0v0 变成了 0 π 0）。如果第 i 片草坪覆盖了 x 点上的 0 π 0 ($l[i] \leq x \leq r[i]$)，那么这只 0 π 0 就可以吃掉这片草坪里的草。每一片草坪的草需要且只能被一只 0 π 0 吃掉。如果一片草坪覆盖了多只 0 π 0，Sullivan 可以选择任意一只去吃草。

但是，0 π 0 吃草是有代价的，对于第 i 片草坪，假如吃草的 0 π 0 位于 x 点上，代价为 $\text{abs}((x-l[i])-(r[i]-x))$ ，即 0 π 0 到草坪两端距离之差。

现在，Sullivan 想知道：

1. 最少需要放几只 0v0？
2. 在放最少只数的 0v0 情况下，代价最小是多少？

3.2 输入格式:

第一行两个正整数 n ， t 。

第二行到第 $n+1$ 行，第 $i+1$ 行两个正整数 $l[i]$ ， $r[i]$ 。

3.3 输出格式

第一行一个非负整数，为最小的 0v0 数量。

如果 $t=0$ ，没有第二行输出；如果 $t=1$ ，第二行输出在放最少只数的 0v0 情况的最小代价。

3.4 样例输入

```
3 1
1 11
2 4
5 7
```

3.5 样例输出

2
0

3.6 样例解释

在 3 上与 6 上各放一只 0v0 即可

3.7 数据范围

20% $n, l[i], r[i] \leq 3000 \quad t=0$
30% $n, l[i], r[i] \leq 300000 \quad t=0$
20% $n, l[i], r[i] \leq 3000 \quad t=1$
30% $n, l[i], r[i] \leq 300000 \quad t=1$