翻转 (rev)

> 有一个排列 a ，初始为 $1,2,\cdots,n$ 。
> 有人在上面进行了 k 次操作，每次选择了 $1\le l\le r\le n$ ，把 $a_l,\cdots,a_r$ 翻转。
> 给定最终的排列，你需要同样用 k 次操作还原为初始状态，或判断无解（即他在要你）。
>
> $n\le 10^5,k\le 3$。

直接枚举 $[l_i,r_i]$ 一共 $2k$ 个数的大小关系（要考虑重合和中间剩余的数），一共最多 $2k+1$ 个数，爆搜一下，然后分别 $O(n)$ 判断。

染色

> 有一棵 n 个点的以 1 为根的有根树，保证每个点的儿子数量是 0 或奇数。
> 每个叶子节点初始可能是黑色或白色，也可能没有颜色。
> Alice 和 Bob 轮流操作，Alice 先手。Alice 每次操作选择一个没有颜色的叶子节点染成黑色，而 Bob 每次操作则是选择一个没有颜色的叶子节点染成白色。
> 叶子全都有颜色之后，从下往上推，非叶子节点的颜色是其儿子颜色的众数。
> Alice 希望让 1 最终成为黑色，而 Bob 希望让 1 成为白色。问谁能获胜。若 Alice 获胜，则还需要求出她第一次可以在哪些叶子操作。

对于一个挂着一些叶子结点的点，如果叶子里 `A` 更多，那么它肯定已经被 `A` 占领了（因为如果 `B` 占一个这个点下方的点，那么 `A` 可以跟着操作）；如果一样多，那就是无主。树形 dp 后，去掉所有有主人的点，剩下的无主点一定是包含根的一个连通块。先手必胜，当且仅当根是无主的。

加边

> 有一条链 $1-2-\cdots-n$ 。
> 进行 m 次操作，每次给定 u_i,v_i ，新增一条边 (u_i,v_i) 。保证操作结束时图中不存在重边自环。
> 每次操作之后，设当前图中共有 k 条边，你需要求出在 $\binom{k}{2}$ 种删除两条边的方案中有几种能使得图不连通。

首先有那个经典的三连通性判定，然后考虑怎么做。一种办法是转成平面覆盖，要求求出一个最小值的桶。可以找到最小的矩形，然后把其它矩形往几个方向递归，把完全包含递归矩形的矩形全部合并，复杂度应该就是 n 的一点几次方。

另一种想法是，我们维护一个边被哪些边覆盖，这构成一个字符串，我们将其后缀排序，然后变为一个求 lcp 的和的问题。这个才是真正能过的，它充分应用了本题的性质。

Euclidean Algorithm

You sit an exam and keep wondering how to calculate the greatest common divisor. With utmost effort you manage to recall that $\gcd(x, y)$ is the greatest integer $d \geq 1$ that divides both x and y. You vaguely remember something along these lines having appeared at one of the lectures. Alas, even if your mind has managed to register any shreds of information during the class, they are now long gone, displaced by the adventures of Gebyte the Witcher as well as the newest episodes of Peaky Byters.

You start with x and y on the blackboard... then you compute some results of subtraction of the previously written numbers...? You try to chase away an unpleasant thought that it is going to end up exactly like the median problem last year... Or was it just one of your nightmares?

Based on your foggy memories from the lecture you came up with the following algorithm. You start with numbers x and y written on the blackboard. In one step you can choose any two numbers a and b that are already written on the blackboard, and write down $c = 2a - b$, conditional upon $c > 0$. The result for the pair (x, y) is the lowest number that can be written on the blackboard after repeating the above procedure some number of times (maybe zero).

After checking a few test cases, your algorithm looked promising. For example, for a pair $(10, 14)$, one of the numbers that you can write down in the first step is 6 (because $2 \cdot 10 - 14 = 6$), while in the next step you can write 2 (because $2 \cdot 6 - 10 = 2$). On the other hand, one can prove that it is impossible to obtain 1, so the algorithm's answer is correct. Unfortunately, for $(10, 16)$ the lowest number you managed to get was 4, while the correct answer is 2. Something is amiss...

For a given n determine the number of pairs (x, y) $(1 \leq x < y \leq n)$ for which your algorithm returns the correct value of $\gcd(x, y)$ [1].

Observe that this task has a low memory limit – 8MB.

- $z \leq 3000$, $n \leq 10^6$
- $z = 30$, $n \leq 10^9$
- $z = 3$, $n \leq 10^{11}$

设 $x<y$，$y=x+k$，那么我们发现可以构造出所有：

$$x + zk \mid x + zk > 0 \tag{1}$$

也就是说，可以构造出 gcd 完全等价于 $y-x|y-\gcd(x,y)$，所以我们直接这么数：

$$\begin{aligned}
\text{Ans} &= \sum_{x<y} [y - x \mid y - \gcd(x,y)] \\
&= \sum_{d} \sum_{xd \leq n, yd \leq n} [y - x \mid y - 1] \cdot [\gcd(x,y) = 1] \\
&= \sum_{d} \sum_{xd \leq n, yd \leq n} [y - x \mid y - 1] \\
&= \sum_{d} \sum_{yd \leq n} d(y-1) \\
&= \sum_{y \leq n} d(y-1) \cdot \left\lfloor \frac{n}{y} \right\rfloor
\end{aligned}$$

实际上，更为本质而简洁的推法：

$$x = (k-1)d + 1, y = kd + 1 \tag{2}$$

$$\text{Ans} = \sum_{d} \sum_{k} \left\lfloor \frac{n}{kd+1} \right\rfloor$$

然后对于整除模组上的每一段分别算 $\sum d$，可以用 $O\left(n^{\frac{1}{3}}\log n\right)$ 求。

Flower Garden

> In front of the Bytingham Palace there is a beautiful garden. Every year masses of travellers wish to see this wonder with their own eyes. The King Intles III has been investing for years mainly in its length, so as many as $3n$ flowers can be planted in a row.
>
> The current gardener, who invested a lot of energy in this majestic subject of the royal pride, has recently decided to retire early, even before turning forty years old. You have just arrived at the palace to take over his role and although the glimpse at the face of your predecessor made you question your abilities to estimate age of other people, you eagerly accepted the job offer. Now the first task awaits you!
>
> King Intles has decided that two kinds of flowers will be planted in the garden this year: violets and roses. They have to conform to the scheme, which is defined by the multi-page royal decree. The first page states as follows:
>
> - *Flower beds for violets and roses are numbered from* 1 *to* $3n$
>
> The next pages state:
>
> - *At least one of the following conditions must be satisfied:*
>   - *All flowers planted on beds from* a_i *to* b_i *inclusive must be roses.*
>   - *All flowers planted on beds from* c_i *to* d_i *inclusive must be violets.*
>
> You are astonished to find q pages with almost the same instructions, differing only by values of a_i, b_i, c_i, d_i. It is not that bad so far, but there is also one terrifying statement on the last page:
>
> - *There are* $2n$ *roses and* $2n$ *violets available.*
>
> Suddenly you remember the face of the gardener you saw when coming to the royal palace and you start regretting your decision. Is this task even solvable? Find an assignment of the flowers that meets the conditions, or determine that it does not exist (and start considering how to avoid the king's fury).
>
> ### Input
>
> The first line of input contains the number of test cases z ($1 \le z \le 10^5$). The descriptions of the test cases follow.
>
> The first line of each test case contains two integers n and q ($1 \le n \le 33\,333, 1 \le q \le 10^5$).
>
> Next q lines contain the description of the royal decree. Each i-th line consists of four numbers a_i, b_i, c_i, d_i ($1 \le a_i \le b_i \le 3n, 1 \le c_i \le d_i \le 3n$) with the meaning described in the problem statement.
>
> Sum of n in all test cases does not exceed $333\,333$. Sum of q in all test cases does not exceed 10^6.

首先我们发现 $[a_i,b_i]$ 中有任何一个 `F`，都能推出 $[c_i,d_i]$ 全都是 `F`，所以我们线段树优化建图一下，然后缩点，我们要求一个闭合子图，带权和在 $[n,2n]$ 间。对于大于 n 的强连通分量，枚举其选不选；对于其它的分量，我们按照拓扑序慢慢选，直到和大于 n 为止，显然不会过头