CF538G Berserk Robot

题目大意

有一个机器人，从原点出发，每秒可以向上/下/左/右移动恰好一步（不能不移动），已知这个运动具有周期性，周期长度为l（也就是说第i秒的移动方向与第$i+l$秒的相同）。给
器人在n个时刻t_i的位置(x_i, y_i)，判定是否有解，如果有解的话还要求出一种方案。

数据范围

$1 \le n \le 2 \times 10^5, 1 \le l \le 2 \times 10^6$

$1 \le t_i \le 10^{18}, -10^{18} \le x_i, y_i \le 10^{18}$

解题过程

首先可以考虑在什么条件下从一个点P出发走k步可以到达点Q。不难发现，P和Q的曼哈顿距离d要$\le k$。并且，当k为偶数时，P的坐标的奇偶性必须和Q的坐标的奇偶性相同，
奇数时奇偶性必须不同。实际上，反过来也成立。这里给出一个构造：从P出发走d步到达点Q，则根据奇偶性条件剩余步数一定为偶数，所以只需要在点Q反复横跳即可。所以，我
以先判掉输入的奇偶性，这样就只用考虑距离了。

然后可以发现在t_i时刻走到(x_i, y_i)的过程实际上是$\left\lfloor \frac{t_i}{l} \right\rfloor$个完整的周期再加上长度为$t_i \bmod l$的零散段拼接起来的。还可以注意到，若$i$对应的零散段长度不大于$j$对应的零散段长度
在已知t_i时刻的位置为(x_i, y_i)的条件下，条件"t_j时刻的位置为(x_j, y_j)"等价于将数量为j的周期数与i的周期数之差（可能为负数或0）的完整的周期和从$t_i \bmod l$到$t_j \bmod l$的零散段拼接
后的结果等于$(x_j - x_i, y_j - y_i)$。所以可以按$t_i \bmod l$排序后差分，这样不同的给定位置对应的零散段没有重叠，因此互相独立，所以可以用第一段的结论。设一个完整的周期会机器
$v = (x, y)$。可以发现，转化后，每个条件都形如："$k_i \cdot v$和(x'_i, y'_i)之间的曼哈顿距离不超过d_i"，其中k_i, x'_i, y'_i, d_i可以通过计算得出。

$k_i = 0$的情况是很简单的，当$k_i \ne 0$时，根据曼哈顿距离的基本性质，可以发现满足第i个条件的v恰好在一个倾斜45度的正方形范围里。因此，我们只需要对所有的正方形求交
如果交里包含奇偶性符合要求的整点则有解，否则无解。然后如果有解则可以根据v的值构造每一个零散段的操作。

CF674D Bearish Fanpages

题目大意

有一个顶点带权的基环内向树森林。设点i的权值为t_i，度数为d_i，则它会给它的每个孩子和父亲带来$\left\lfloor \frac{t_i}{d_i+1} \right\rfloor$的贡献，同时给自己带来$t_i - d_i \times \left\lfloor \frac{t_i}{d_i+1} \right\rfloor$的贡献。每个点的答案为所
给它带来的贡献之和。要支持三种操作：

- 修改一个点的父亲
- 输出一个点的答案
- 输出所有点的答案的最小值和最大值

数据范围

设操作次数为q

$3 \le n \le 10^5, 1 \le q \le 10^5$

修改操作不超过50000个。

解题过程

容易发现，每个点的答案只与它自己的权值和度数，它的孩子的权值和度数，和它的父亲的权值和度数有关。

考虑修改操作，可以发现原来的父亲度数会减小1，新的父亲度数会增加1，其他点度数不变。这样，原来的和新的父亲的其他孩子由于父亲度数变化也会受到影响，因此直接维
案的话时间复杂度会有问题。

考虑树上邻域类问题的一个常见技巧：将一个点所有孩子的信息在这个点一起维护。可以发现，原来的父亲的其他孩子由于父亲度数变化所受到的影响是相等的，因此可以用全
标记维护，新的父亲的其他孩子同理。

那有哪些点的答案可能会发生变化呢？根据第一段，并注意到权值不会改变，而度数发生改变的节点为原来的和新的父亲，邻域结构发生改变的节点为它自己和两个父亲，因此
可能会改变的点只有：它自己，原来的父亲，新的父亲，原来的父亲的孩子，新的父亲的孩子，原来的父亲的父亲，新的父亲的父亲。其中，XXX的孩子我们已经考虑过了，而剩下
只有$O(1)$个，可以直接维护。

考虑需要维护哪些信息：为了求出所有点的答案的最小值和最大值，我们可以对每个点用一个set维护它的孩子的答案的最小值和最大值，然后再维护一个全局的set存放每个点的
的答案的最小值和最大值，这样就可以回答第三种询问了，当一个点发生变化时我们要先更新它父亲的set（因为我们把所有孩子的信息在这个点一起维护），然后如有需要再更新全
set，而对一个点所有孩子的整体操作可以直接打上标记然后更新全局的set。

ARC101F Robots and Exits

题目大意

直线上有n个小球和m个洞。你每次可以将所有还在直线上的小球同时向左或向右移动一个单位的距离。当一个小球的位置和洞相同时，它会掉进洞里并从直线上消失。求出所有
球都掉进洞里的方案数。这里我们只关心最终每个球掉进了哪一个洞。也就是说，两种方案是不同的，当且仅当某个小球在这两种方案里掉进了不同的洞。

数据范围

$1 \le n, m \le 10^5$

坐标均为正整数且不超过10^9

解题过程

容易发现一些结论：

- 最左边的洞的左边的小球和最右边的洞的右边的小球是不会影响答案的
- 只有往左移动的距离比之前往左移动的距离的最大值还要大的操作和往右移动的距离比之前往右移动的距离的最大值还要大的操作才有用，因此可以直接考虑向左移动的的最大值和向右移动的距离的最大值
- 对某个小球来说，它会进哪一个洞的只与“向左移动的距离的最大值达到小球的位置到它左边最近的洞的距离”和“向右移动的距离的最大值达到小球的位置到它右边最近的距离”这两个事件哪一个先发生有关

考虑将问题放到二维平面上，那么小球就对应着一个点，你的移动对应着一条格路。而结论二就是每一步只能向上或向右走，结论三就是两个方案不同当且仅当对于某个小球对点，一条路径从点的左边向上走达到点所在的水平直线，而另一条路径从点的下面向右走达到点所在的竖直直线。

考虑如何计数：直接统计格路数量而不考虑不同方案的要求是会重复计算的，因此可以考虑将对应相同方案的不同格路标准化为其中的某个格路，再统计标准格路的数量。比如如果不会经过小球对应的点的水平直线和竖直直线，那么先向上走一步再向右走一步和先向右走一步再向上走一步是一样的，所以我们可以考虑要求不到向上的ddl就一直向右走，上走需要的距离（一到目标平直线就向右走）。可以发现，通过调整，所有格路都可以化为这种形式，且不同方案对应不同标准形式，相同方案对应相同标准形式。

这样，我们就可以从左向右维护当前标准形式的数量。当要到一个点的横坐标（ddl）时需要将纵坐标小于点的纵坐标的标准形式转移到点的纵坐标，可以用树状数组维护。