Sky 要玩棋子

20分

为了两两抵消，必须选数目相同数量的红黑色棋子。

当 $m=1$ 时，只需要选一个红色棋子就好了，所以答案是 n

40分

用 C_i^j 表示从 i 个不同的物体种选择 j 个物体的方案数

那么枚举选择的棋子的数量，答案就是

$$\sum_{i=1}^{min(n,m)} C_n^i C_m^i$$

通过递推公式 $C_i^j=C_{i-1}^{j-1}+C_{i-1}^j$，可以在 $O(n^2)$ 的时间内预处理出所有的 C 值，然后每次 $O(n)$ 计算上述式子的值即可。

100分

假设我要选择 i 个红色棋子，那么相应的我就要选 i 个黑色棋子。

也就是说，没被选中的黑色棋子有 $m-i$ 个。

也就是说，“选 i 个红色棋子再选 i 个黑色棋子”和“选 i 个红色棋子再选 $m-i$ 个黑色棋子”之间建立了一一对应关系。

而选 i 个红色棋子，再选 $m-i$ 个黑色棋子，其实就是在全部的棋子中选出 m 个，由于不能一个都不选，所以答案还要减去 1

所以答案就是 C_{n+m}^m-1

预处理阶乘，每次 $O(1)$ 计算即可

Sky 要玩集合

30分

$n, m, a \le 1000$

按题意暴力模拟即可。

时间复杂度: $O(nm)$

60分

首先对于前面 30% 的数据进行暴力

然后对于 30% 的数据, l, r 随机。

因为 l, r 随机，每次更新最长连续数字的概率都不是特别高，答案不会太大。

我们建出一颗线段树，向区间插入那样，一个操作分解成 log 个区间操作，对于线段树上的节点，操作就只剩下了插入，那么答案不会太大我们每次插入时用一个 set 维护当前集合，然后对先后暴力查找元素是否存在。

100分

对于 100% 的数据, $n, m \le 100000, 1 \le a \le 10^9$

考虑扫描线

从左往右计算每个集合的答案。

那么每次就是插入一个数或者删除一个数，用线段树维护就好了。

时间复杂度： $O(n \log n)$

Sky 要上树

30%

每次查询的时候递归求解答案，记忆化一下，会计算的点对个数显然是 n^2 级别的，故单次查询的复杂度是 $O(n^2)$，总时间复杂度 $O(qn^2)$

100%

考虑一条链上每一条边的贡献，手动打个表格看看(第 i 行第 j 列的数字代表长度为 $i + j - 1$ 的序列中第 j 个元素会被计算多少次):

```
1 1 2 4 8
1 3 8 20
2 8 26
4 20
8
```

容易发现第 i 行 j 列的数 $V(i,j)=\frac{\sum\limits_{x\le i,y\le j} V(x,y)}{2}$

也很好理解这个式子，一个可能包含该元素在某个位置的序列都会出现在左上角，那么就可以直接统计前缀和计算

查询的时候暴力向上跳到 lca 然后统计答案即可，时间复杂度 $O(n^2+qn)$

Sky 要搞序列

30%

手算即可。

50%

发现忽略 $a_{m+1}=0$ 后数字不重复使用，于是考虑状态压缩动态规划：设 $f_{s,i}$ 表示前若干位使用了集合 s 中的数，且最后使用的数为 i 的方案数，则有初始态 $f_{\{0\},0}=1$ 与状态转移方程：

$$f_{s,i}\leftarrow \sum_{j\in s\setminus\{i\},(2j\equiv i \pmod m)\vee(2j+1\equiv i \pmod m)} f_{s\setminus\{i\},j}$$

时空复杂度 $\mathcal{O}(m2^m)$。

80%

特判 $m=1$，这种情况比较特殊。

观察到如果 m 为除了 1 以外的奇数，那么答案为 0。证明也比较简单：由于不能有 $2a_m\equiv a_{m+1}\equiv 0 \pmod m \implies a_m=0$，因此只能有 $2a_m+1\equiv a_{m+1}\equiv 0 \pmod m \implies a_m=\frac{m-1}{2}$。设 $a_k=m-1$，由于不能有 $2a_{k-1}+1\equiv a_k\equiv m-1 \pmod m \implies a_{k-1}=m-1$，因此只能有 $2a_{k-1}\equiv a_k\equiv m-1 \pmod m \implies a_{k-1}=\frac{m-1}{2}$。而明显 $m-1\neq 0$，因此 $k-1\neq m$，两者不能同时满足。故 m 为除了 1 以外的奇数时答案为 0。

现在考虑 m 为偶数时的情况：在模 m 意义下对于任意的 x， x 和 $x+\frac{m}{2}$ 的下一个数都只能是 $2x$ 或者 $2x+1$，而这两个数是不同的，因此若 x 的下一个数是 $2x$（或者 $2x+1$），则 $x+\frac{m}{2}$ 的下一个数只能是 $2x+1$（或者 $2x$），反之亦然。因此我们只需要决定其中一半的数的下一个数，另一半数就可以直接构造且只有唯一的方案。

因此考虑对 $0,1,2,\cdots,\frac{m}{2}-1$ 这些数形成的集合作动态规划，设 f_s 表示当前使用了集合 s 中的数的方案数，则有初始态 $f_{\{0\}}=1$ 与状态转移方程：

$$f_s \leftarrow \sum_{i\in s,\exists j\in s\backslash\{i\},(2j\equiv i\pmod m)\vee(2j+1\equiv i\pmod m))} f_{s\backslash\{i\}}$$

时间复杂度 $\mathcal{O}(m2^{\frac{m}{2}})$，空间复杂度 $\mathcal{O}(2^{\frac{m}{2}})$。

100%

发现上面这个动态规划式子实质上就是在求一个有向图的有根外向生成树个数，因此可以使用矩阵树定理优化。具体地，连边 $i\to(2i\mod m)$， $i\to(2i+1\mod m)$，边权均为 1。时间复杂度 $\mathcal{O}(m^3)$，空间复杂度 $\mathcal{O}(m^2)$。由于实现时时间有约 $\frac{1}{16}$ 的常数，因此可以通过。