Problem A - Light

难度：1

签到题，注意到我们每次操作都是连续k个，所以对于每个可能的左端点只可能被翻转一次，所以我们从左到右枚举，遇到必须翻的灯就翻一次，同时维护一下对右边的影响就好了，时间复杂度$O(n)$

Problem B - Slogan

难度：1

对于每个a 统计一下左边有多少个q 右边有多少个q 乘起来即可

Problem C - An Interesting Game

难度：2

考过$noip$的同学应该知道前面所给的结论是$noip2017$的题目小凯的疑惑

这题是它的一个扩展

我们容易发现，若k这个数不能表示，则k-a,k-b一定是不能表示的

利用所给的样例通过找规律其实就可以发现对于任意

$x = a * b - k1a - k2b(x > 0, k1 > 0, k2 > 0)$也一定是不能表示的

有了这个结论就很容易解决这题了

具体证明可以参见小凯的疑惑这道题

Problem D - Sum of Sets

难度：2

由于 m 次操作之后才询问，我们可以考虑离线算法扫描线。

对于一次操作 L, R, x ，我们在 L 处添加 x ， $R+1$ 处删除 x 。

从 1 到 n 扫描，记录 cnt_x 表示当前 x 的出现次数，和当前答案 sum ，我们不难想到：

- 如果添加 x 时 $cnt_x = 0$ ，那么说明 x 是新出现的，sum 加上 x
- 如果删除 x 时 $cnt_x = 1$ ，那么说明 x 已经不存在了，sum 减去 x

处理到 i 时的 sum 就是集合 i 的和。

Problem E - Magic Words

难度： 3

V4yne有n句魔咒，第 i 句魔咒可以用 ai 次，每句魔咒包含 xi 种元素。

有m个怪兽，每个怪兽只有一个元素。

每一种元素都有一个 id 值来表示，每次使用魔咒可以对一个和这个魔咒包含的某个元素相同的怪兽造成1HP的伤害。

要求得最大的对怪兽的伤害总值，贪心显然不能解决。考虑对元素来建图利用网络流解决。

建立一个超级源点为0号点，1~n号点代表每一句魔咒，从超级源点到 i 号点连边，最大流量设置为 ai，最大流量意味该魔咒最多只能使用 ai 次。

第 n个点后面的 n+m 个点代表每一个元素属性，如果第 i 个魔咒有 id 号元素，就将 i 号点连一条边到 n+id ，最大流量为inf。

设置一个超级汇点为2*n+m+1。

设置一个数组 b[N]。

再遍历m个怪兽，如果第 k 个怪兽的元素是 id，b[id] 就加上第k个怪兽的血量。

此时 b[id] 表示所有元素为 id 的怪兽的总血量。

最后从1~n+m遍历 b 数组，假设当前遍历到的是第 id 号元素从第 id 个元素代表的点（n+id）向超级汇点连一条边，最大流量设为 b[id]。

最后从超级源点向超级汇点跑网络流，最大流即为V4yne最多能对所有怪兽造成的总伤害，如果最大值等于所有怪兽血量的和，那么V4yne获胜。如果不相等，输出最大流即可。

NOTE：题目只给了1s的时限，如果是将每个魔咒和每个怪兽进行拆点来跑网络流会超时，题解解法是点数与边数都更少，效率更优秀的建图方法。

Problem F - TangTang's Question

难度：3

我们先考虑如果是问题是离线如何处理。

扫描序列，用树状数组维护，扫描到当前点时，假设上一次出现的位置是x,当前点是i,在x这个位置$+(i-x)$.

回答以当前点为右端点的询问，答案为询问对应的区间和。

在线时使用主席树。

对当前点进行建树时，继承前一个点主席树的信息，并且对x这个位置$+(i-x)$。

每次询问时，在r_i对应的主席树上进行区间查询即可。

Problem G - Play on the Graph

难度：2

> 一张 n 个点的无向完全图，从 1 号点出发，每秒随机选择当前点的一条出边移动
>
> q 次询问，每次询问在第 t_i 秒在 1 号点的概率，答案对 998244353 取膜。

设 $f[t]$ 表示在 t 秒时在 1 号点的概率，则有：

$$f[0]=1,\ f[t+1]=(1-f[t])*\frac{1}{n-1}$$

直接矩阵快速幂加速递推即可。

此外，如果您愿意推一推式子，配方 + 等比数列化简即可得：

$$p=\frac{1}{n-1},\ f[t]=\frac{p+(-p)^t}{1+p}$$

直接快速幂计算即可。

Problem H - Escape From The Secret Room

难度：3

只要能算出根节点输出是1的概率$prob$，那么期望次数就是$\frac{1}{prob}$，P可以用总方案数除以输出是1的方案数得到，而总方案数就是2^{leaves}，因为每个叶子节点有两种输入的可能。

接下来考虑如何计算输出是1的方案数，整个电路是个树状结构，且所有状态都是从叶子节点往根节点上转移的，考虑动态规划：$dp[i][j](i \in [1,n], j \in \{0,1\})$表示在$i$号节点，输出是0或者1的方案数，由于节点类型有四种，自下而上分别对四种节点进行转移：

为了便于描述，记$son(u)$表示u的儿子节点集合，u为当前处理的节点，$|S|$表示集合S的元素个数

1. controller(叶子节点)，$tp=-1$
   显然叶子节点满足$dp[u][0]=dp[u][1]=1$
2. AND gate(与门)，$tp=0$
   只有当输入全部是1的情况下，输出才是1，也即$dp[u][1]=\prod_{v\in son(u)} dp[v][1]$，输入的所有可能性为$all=\prod_{v\in son(u)}(dp[v][0]+dp[v][1])$，所以$dp[u][0]=all-dp[u][1]$
3. OR gate(或门)，$tp=1$
   和与门相反，只有当输入全部是0的情况下，输出才是0，所以$dp[u][0]=\prod_{v\in son(u)} dp[v][0]$，输入的所有可能性为$all=\prod_{v\in son(u)}(dp[v][0]+dp[v][1])$，所以$dp[u][1]=all-dp[u][0]$
4. XOR gate(异或门)，$tp=2$
   只需要按输入的1的数量是奇数个还是偶数个分类即可，可以用一个简单的dp来做，$f[i][j](i \in [1,size(son(u))], j \in \{0,1\})$表示当前考虑到第$i$个子节点，输入1的数量是奇数/偶数个的方案数，初始状态为$f[0][0]=1, f[0][1]=0$，转移方程为：
   $f[i][1]=f[i-1][0]\times dp[v][1]+f[i-1][1]\times dp[v][0]$
   $f[i][0]=f[i-1][0]\times dp[v][0]+f[i-1][1]\times dp[v][1]$
   其中v表示u的第i个子节点
   最后$dp[u][0]=f[|\ son(u)\ |][0], dp[u][1]=f[|\ son(u)\ |][1]$

最后$dp[1][1]$就是输出根节点输出是1的方案数

Problem I - GieGie and Cube

难度：1

题意：求$(\sum_{i=L}^{R} i^3)^3$

外面这个三次方可以不用管，只要求出里面的东西，最后乘一下取模就好了。里面这个东西可以用前缀和的思想，转化为$\sum_{i=1}^{R} i^3-\sum_{i=1}^{L-1} i^3$，那么我们只需要知道$1^3+2^3+3^3+.....+x^3$这个东西的通项公式就好了。相信大家都知道自然数列的求和公式是$\frac{x*(x+1)}{2}$，赛场上也不必尽力回想立方的求和公式，手推四项可得前四个立方和：1，9，36，100。不难发现，这四项等价于1^2，3^2，6^2，10^2。这时《对数学的敏感》使你发现了这四项的底数分别是1,1+2,1+2+3,1+2+3+4。

于是猜到$\sum_{i=1}^{x} i^3 = (\frac{x*(x+1)}{2})^2$

当然通过一些数学上的操作也可以得到这个公式，但出于结论比较容易猜所以这题定档为签到题。

但是猜到结论也不要太兴奋，由于模数是1e10+7（题目是个提示！），模数范围的数字相乘可能会爆long long，所以要用龟速乘或int128。

Problem J - Push-ups Triggered by a Basketball

难度：1

按照题意模拟即可，枚举要篡改的位置，然后按照题意暴力计算答案取最优即可

时间复杂度$O(n^2)$

Problem K - Quaternions

难度： 3

要找到四个数的乘积是平方数，显然就是这个乘积不断去除平方因子之后是1。
然后把四元组分成前两个和后两个两部分，令$f(x,y) = xy/gcd(x,y)$,那么我们需要找的就是有多少四元组满足$f(a_i,a_j) = f(a_k,a_w)$。
我们可以先枚举第三个数，于此同时我们可以顺便在$O(n^2)$时间内预处理好前$i-1$个数能凑出的所有数对，当前枚举到a_i时，在枚举第四个数用来询问。
这样就能做到$O(n^2logn)$的效率了,此时已经能过了。
其实gcd是可以预处理的，所有可以进一步优化到$O(n^2)$。