Contest 9 by Abracadabra

[Iori] Bulbasaur and Rare Candies

- 有N堆糖，每堆有绿蓝两色。A只能拿绿色，B只能拿蓝色，每次至少从一堆里拿一颗。拿到最后一颗的一方获胜。A先手，问谁获胜。

- 显然要想坚持到最后，每次都要拿最少的糖。
- 所以每次拿一颗就好了...
- 因为A是先手，所以绿色糖总量要严格大于蓝色才能获胜。否则B获胜。

[Miki] The Biggest Happiness

- 按题目的公式找出最大的happiness. 只有两条路。

- 读懂题可能有点费劲（很抱歉描述有很多不清楚的地方，开始的sample也出了很多问题）……方法很简单。
- 根据图可以直接考虑两种情况：一个是在第一个岔路拐弯 一个是不拐弯。
- 第一条路肯定不会选第二个shop，所以判断一下在第一个shop买冰棒是否划算。
- H1=-2*m1+5*(5*c1-11*c3)
- H2=5*11*c3——这里因为开始样例的bug很多学长写的4过不去。

[Miki] The Biggest Happiness

- 第二条路也同样只可能选第一个全家，快递肯定要取。
- H3=-2*m2+5*(3*c2-13*c3)+3*s
- H4=5*13*c3+3*s
- 比较得出最大的H.
- 为了防止拿着冰棍到处溜达增加happiness的情况，拿了以后只能直接回寝，不然可能∞了。
- 只有两种走路的方法因为不能掉头。

[Chihaya] Run Faster than any Western Reporter

- 将2n个数均分两组，使得其中一组加上（或者乘以）一个常数与另一组一一对应。求这个常数的最小值。

- 从小到大排序后，枚举和最小值对应的数（即枚举q或d），把这两个数标记已使用，然后从小到大扫描数组，不断找到未标记的数，找和他是否存在对应的数。

[Chihaya] Run Faster than any Western Reporter

- 怎么找：
- 1.这个数可以二分查找，由于可能出现很多重复数据，二分后还要加一些特判。
- 2.由枚举的性质，当前未标记数对应的数必定是递增的。维护对应数的指针即可。由于这个指针是递增的，扫描时间复杂度O(n)，外层枚举q/d也是O(n)。总时间复杂度O(n^2).

[Chihaya] Run Faster than any Western Reporter

- 虽然在forum上指出q可能为1，但是事实上并没有这样的数据。然而，由于允许重复数，在枚举过程中q可能等于1（或d等于0）。
- 这题并没有要卡大家的意思...数据比较弱O(n^3)都能过...很多学长一开始wa了...
- 一个原因：指针维护时，注意至少应当从当前未标记数的下一个数开始。因为可能当前枚举的q等于1（或d等于0），如果不是从当前未标记数的后一个数开始，指针会指向这个未标记数自己。
- 举例：q=3，n=3　　1 1 1 3 3 3
- 第一轮枚举1（第二位）作为1（第一位）的对应数。得到q等于1，此时check时，如果指针不从未标记数下一个数开始，程序会认为q=1也是可行的，从而出错。

[Azusa] N-Divide

- 将n拆分为k个数，k个数最大公约数为1.

- 考虑更简单的问题：将n拆分为k个数，不要求总的公约数为1.
- 答案：C(n-1,k-1)
- 这时候包含了很多gcd(a1,a2...)=2,3,...的情况，要进行容斥。

[Azusa] N-Divide

- 莫比乌斯反演 $\mu(N)=\begin{cases}1 & n=1\\ -1^k & n=p_1p_2\cdots p_k,\ p_i\text{为不相同的质数}\\ 0 & others\end{cases}$
- 显然，$\sum_{d|n}\mu(d)=[n==1]$
- 假设我们要求F(k)，令$G(d)=\sum_{d|k}F(k)$
- 根据容斥原理，猜测$F(k)=\sum_{d|p}G(p)\times Q(p)$.其中，Q(p)为系数。
- 莫比乌斯反演告诉我们，$F(k)=\sum_{k|p}\mu\left(\left(\frac{p}{k}\right)\times G(p)\right)$

[Azusa] N-Divide

$$F(k) = \sum_{k|p} \mu\left(\frac{p}{k}\right) * G(p)$$

$$= \sum_{k|p} \mu\left(\frac{p}{k}\right) \sum_{p|q} F(q)$$

$$= \sum_{k|q} F(q) \sum_{p|\frac{q}{k}} \mu(p)$$

$$= \sum_{k|q} F(q)[\frac{q}{k} == 1]$$

$$= F(k)$$

gcd(a1,a2....ak)=p)

$$G(k) = \sum_{k|q} F(q)$$

$$F(q) = \sum_{q|d} \mu(\frac{d}{q}) G(d)$$

$$F(1) = \sum_{d=1}^{n} \mu(d) G(d)$$

[Yukiho] Bulbasaur and Exciting Game

- 设 $X_i = a_i X_{i-1} + B_i \qquad X_0 = 1$ ，有三种操作：
- 1、修改一对 a_i, b_i
- 2、把L ≤ i ≤ R范围内的 a_i, b_i 取自然对数并向下取整。如果原来是0则不变。
- 3、询问Xi的值。

[Yukiho] Bulbasaur and Exciting Game

- 先考虑一下没有操作2应该怎么做。
- 记 $M_i=\begin{bmatrix} a_i & b_i \\ 0 & 1 \end{bmatrix}$ ，$\begin{bmatrix} X_i \\ 1 \end{bmatrix}=M_iM_{i-1}...M_1\begin{bmatrix} X_0 \\ 1 \end{bmatrix}$.
- 那我们用线段树来维护矩阵的乘积就可以了。
- 还可以注意到：矩阵相乘以后，第二行总是0 1，所以只需要维护上面两个数就可以了。
- 最后答案就是queryA(1,i)+queryB(1,i)

[Yukiho] Bulbasaur and Exciting Game

- 再来考虑一下操作2.
- 我们发现，操作1是单点修改，而操作2是区间ln.
- 一个区间被ln 4~5次之后，就会全部变成0，而想要把它变回非0的区间，就需要“区间长度次”操作1.
- 也就是说，有效的大范围ln操作不会很多，完全可以暴力处理。
- 我们记isZero标志表示一个区间里所有的a和b是不是均为0，ln操作时，碰到isZero为真的区间，就可以不用往下做了。ln操作均摊后的复杂度是每次log的。
- 所以总复杂度O(nlogn).
- 暴力出奇迹...

[Hibiki] Carrot Gathering

- 给定一个n个点m条边的无向图，每次询问从某个点出发经过所有权值不超过x的边所能到达点的权值之和。点权可能被修改。
- 强制在线

[Hibiki] Carrot Gathering

- 性质：所有可到达点集合互不相交，只会包含或相离。
- 自底向上合并区间建树，将图转化为树形结构，再根据树上叶子节点（原图中节点，树中意义为大小为1的区间）的DFS序转化为链。
- 查询点和区间的转化可以用树上倍增处理。
- 用树状数组或线段树处理查询。