A. A-SOUL!

Author: ZCQ

从小到大枚举开头即可，如果某个数已经被其他开头的数用过了，那它就不可能成为答案了。

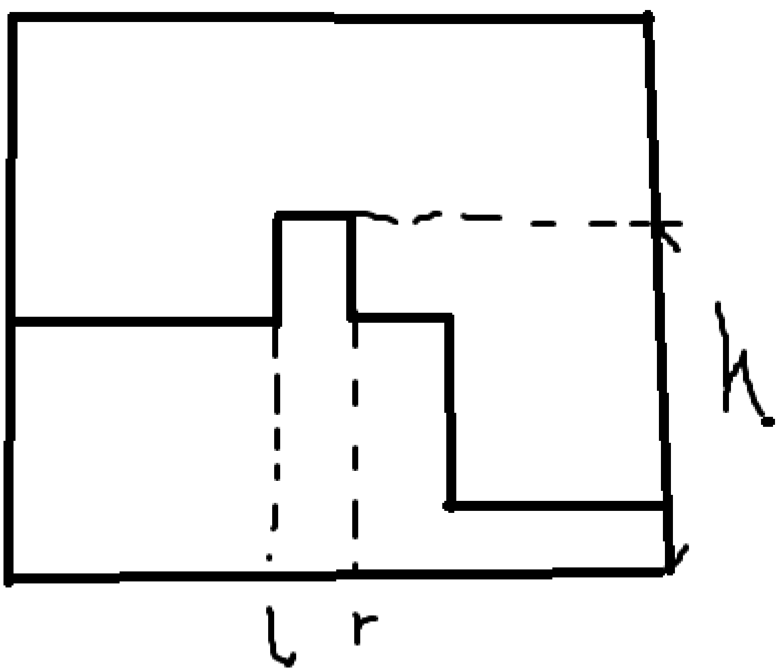
所以每个数只会被访问到一次。

B. Bheith i ngra le

Author: PXA

题意：给一个n列m行的网格图，问符合条件的山型图案有几个。

方法：我们先考虑枚举山顶的位置，即l, r, h



并规定$h_{l-1} < h_l, h_r > h_{r+1}$，因为相等的情况可以通过枚举不同的$l, r$得到，于是问题转化成了：在一个$x$行$y$列的网格图里画一个单调不下降的曲线（显然单调不上升与单调不下降是对称的），可以使用dp转移或者组合数学解决，这里使用组合数学，要求的种数是C_{x+y}^{x}，那么式子就是

$$\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=1}^{m}C_{i+k-2}^{k-1}\times C_{n-j+k-1}^{k-1}$$

但这个式子是$O(n^3)$的，我们需要优化。

发现第一个组合数和j没有关系，所以可以进行变形。

$$\sum_{i=1}^{n}\sum_{k=1}^{m}C_{i+k-2}^{k-1}\times\sum_{j=i}^{n}C_{n-j+k-1}^{k-1}$$

且后面那个式子除了i开始外和i也并没有关系，那么我们就可以预处理

$$num[i][k]=\sum_{j=i}^{n}C_{n-j+k-1}^{k-1}$$

那么最终式子就变成了

$$\sum_{i=1}^{n}\sum_{k=1}^{m}C_{i+k-2}^{k-1}\times num[i][k]$$

（出题人就是因为推到这里忘了加取模了QAQ）

总复杂度：$O(n^2)$

C. Counting Cats!

Author: ZCQ

题意确实没有说清楚，给困惑的大家说声抱歉。

只有形如 cat，ca，c，at，a，t 是有用的。

拿 c 和 at 拼，ca 和 t 拼，剩下的 c，a，t 拼一起即可。

D. Dahno Dahno

Author: RHY

先默认所有的 power 都能拿到。

那么就变成了一个无向图全局最小割的问题了，直接 $O(n^3)$。

E. Experiment Class

Author: MX

总的来说，这题的考点在于简单几何的应用。

总共需要考虑三种情况: 1.都在两条射线形成的角内。2.一个在角外。3.都在角外。

对于第一种情况：在这里由于我们限制了射线在第一象限，两个点也在第一象限，所以这种情况下肯定是直接跨过两条直线的。那么直接将两个点连成一条线即可。

第二种情况：他们的连线一定会经过其中一条射线，那么将内部的点关于我们可以将其关于另一条直线做称点，由于对称点的性质：直线上的点到两个对称点的距离相等。我们可以得到将外面的点和里面的那个的对称点做直线即可。

第三种情况：由第二种情况可以想到两个点都做对称点连起来。问题是，哪个点关于哪条线做对称。我们可以直接两种情况取最小值。

对于以上做法我们可能存在连线与两条射线没有交点的情况，直接特判即可。

那么，最后，只需要对在角内部的点做对称点，原先在外部的点不变，特判没有交点的情况后取较小的方案的输出即可。

F. Faineant Komorebi Loves Game

Author: MX

首先我们可以很快的想到，当k=0的时候显然就是威佐夫博弈，那么复习一下威佐夫的结论，局面(a,b)为必败态的条件是满足，$a = floor((b-a)*\frac{1+\sqrt{5}}{2})$,这个时候我们想想这个题目会不会有**类似的性质**呢。

将所有必败态看成两个数组 (a,b) 并规定$\geq a$，且 a 递增。

首先看到第一个必败态显然可以想到是 $(1,k+2)$ 那么所有除了他的类似 $(1,y)$ 或者 $(x,k+2)$ 的局面都是必胜，因为我们永远可以通过取一堆石子的方式使得局面变成 $(1,k+2)$。换句话说，1 还有 $k+2$ 这两个数字都不会再出现在这两个数组中了。显然可以想到，同样的如果有一个必败态(a,b) 那么 a 还有b这两个数字都不会再出现在这两个数组中,那么可以得出，**这两个数组的并集中的所有元素最多出现一次。**

还有一个比较显然的结论，一个数如果从来没有在之前的必败态中出现他一定会在之后的必败态中出现，且在位置 a 。(l,r) 这个状态存在，且 l，r 之前都没有出现过。那么 $(l-k,r)$ 或者 $(l,r-k)$ 都不可能是必败态也就是说不能通过拿走一堆石子的方式得到必败态。当我们已经寻找到 i 对必败态时，如果我们选择暂时只考虑没有出现的最小的正整数 p，我们考虑 (p,y) 是必胜态的条件:由于 p 从未出现过，所以（$p,y-k$）肯定不是之前的必败态，如果 $(p-k,y)$ 是必败态，最多有 $2i$ 种不同的 y 可以使得符合情况，$(p-k_1,y-k_2)$ 由于$|k_2-k_1| \leq k$所以最多有 $(2*k+1)*i$ 种不同情况，但是形如 (p,y)的状态的大小是正无穷，也就是说一定存在至少一个数 y^* 是当前状态无法转移到的 (p,y^*)。

由以上的证明可以得出已经得到 i 组必败态时下一个a_{i+1}=之前未出现的最小正整数。

由 a 进一步得到，所有正整数都会出现。那么如果得到 a 和 b 之间的关系就可以考虑使用贝蒂定理解方程得到 a 数组的通项公式了。

并且由上述证明过程中可以得到关于$y*$也就b_{i+1}是具体取值的一个猜想:
$b_{i+1} = a_{i+1} + (k+1)*(i+1)$

下面给出证明：

如果前 $i-1$ 组满足 $b_{i-1} = a_{i-1} + (k+1)*(i-1)$ 那么 $(a_i, a_i+(k+1)*i)$ 可以转移到的状态有 $(a_i, a_i+(k+1)*i-x)$。由于 a_i 从未出现，所以不可能是之前的必败态；

$(a_i-x, a_i+(k+1)*i)$若$a_i+(k+1)*i$之前出现过，由于a数组是递增的，且
$a_i+(k+1)*i > a_i$所以它不会出现在a数组中，那么只有可能出现在b数组中，那么有
$a_i+(k+1)*i == a_j+(k+1)*j$当且仅当i==j所以$a_i+(k+1)*i$之前也没有出现过，所以
$(a_i-x, a_i+(k+1)*i)$也不可能是必败；

$(a_i-x1, a_i+(k+1)*i-x2)$设$a_i-x1 == a_j$那么$a_i+(k+1)*i-x2 == a_j+(k+1)*j$时可以转移到必败态则$(k+1)(i-j) == x2-x1$由于$x2-x1 <= k$所以这是不可能实现的条件，则没有可行的必败态可能转移。

$a_n = [\alpha * n], b_n = [\beta * n]$，有 $a_n + (k+1)n = [(\alpha + k + 1) * n] = [\beta * n]$，解方程 $1/(\alpha + k + 1) + 1/\alpha = 1$

得到$\alpha = \frac{(1-k)+\sqrt{(1+k)^2+4}}{2}$那么

$a_i = \lfloor \frac{(1-k)+\sqrt{(1+k)^2+4}}{2} * i \rfloor$

$b_i = \lfloor \frac{(1-k)+\sqrt{(1+k)^2+4}}{2} + (1+k) * i \rfloor$

我们可以通过二分 i 的方式查找 x 是在 a 中还是 b 中，并输出对应的另一个值。需要注意一点精度问题。

G. Gentle Jena II

Author: RHY

如果是绕着原点旋转逆时针旋转 $t°$，很容易想到通过矩阵描述旋转

$$\begin{bmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

绕点 (a, b) 旋转，可以看做先平移 $(-a, -b)$ ，再绕原点旋转，再平移 (a, b)。

可以给矩阵升一维描述仿射变换，那么绕着点 (a, b) 旋转 $t°$，就可以写成

$$\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos t & -\sin t & 0 \\ \sin t & \cos t & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

线段树维护矩阵、$\sum x$ 和 $\sum y$ 即可，注意负数取模的细节。

H. Hearthstone

Author: PXA

（真的会有炉石玩家能忍住不看这题题面吗，不会吧不会吧）

题意：同$TL;DR$后的版本（不想写了）。

方法：因为最后的$F(j,k)$和i无关，因此可以先转换枚举顺序，改为枚举j作为因数在1到n中出现的次数，式子变为了：

$$\sum_{i=1}^{n} \lfloor \frac{n}{i} \rfloor \times F(i,k)$$

那么这个式子就很像可以数论分块，当然，数论分块的前提是我们需要知道$F(i,k)$的前缀和，即：

$$\sum_{i=0}^{n} F(i,k)$$

也就是说现在我们只需要解决这个就可以直接代入数论分块了。

这个式子的枚举顺序还是并不明显，因此我们可以再定义一个函数$f(i,k)$表示第k位是不是i的平衡点，根据定义显然有：

$$F(i,k)=\sum_{j=0}^{k-1}f(i,j)$$

那么式子就变成了：

$$\sum_{i=0}^{n}\sum_{j=0}^{k-1}f(i,j)$$

再变换一下枚举顺序：

$$\sum_{j=0}^{k-1}\sum_{i=0}^{n}f(i,j)$$

我们把后面那块拿出来，根据定义解释就是：求所有不大于n的数中以第j位为平衡点的数的个数。

因此最后那块式子可以用数位dp解决。

总复杂度：$O(\sqrt{n}*k^2*p)$

I. Identical Day

Author: RHY

优先队列维护一个二元组 (l,x) ，权值是 l 长的连续段从平均分成 x 段变成 $x+1$ 段，能给 k 减去多少。

然后贪心的拿最大的即可。

J. JXC&Jesus

Author: PXA

（因为我们把题面进行了一次魔改所以没注意到重要的符号出了一些问题，其实是个很简单的题，深感抱歉）

题意：定义了一个函数$F(x,m)$，把x质因数分解为$a_1^{k_1}a_2^{k_2}\dots a_k^{k_k}$后，

$$F(x,m)=a_1^{\lfloor\frac{k_1}{m}\rfloor}a_2^{k_2}\dots a_k^{k_k}$$

给定n,m,L，求

$$\sum_{i=L+1}^{L+n}F(i,m)$$

方法：因为数据开到10^7，所以大约可以猜到是个线性的做法，又注意到F的变换本质上只变化了**最小质因子**，到这里熟悉线筛的同学们就一定会反映过来这是可以线筛出来的。

我们考虑线性筛状态的转移。

- x是质数，那么当$m=1$时$f(x)=x$，否则$f(x)=1$
- 枚举的质数$pr[j]$不能整除当前的i，那么显然i不是$pr[j]$的因数（废话），换而言之，$pr[j]$是$i*pr[j]$的最小质因子，那么此时$m=1$时$f(i*pr[j])=i*pr[j]$，否则$f(i*pr[j])=i$

发现似乎现在维护的并不足以处理$pr[j]$整除i的情况，我们还需要额外维护$cnt(x)$表示x最小质因子的质数，于是就有了最终版本：

- x是质数，那么当$m=1$时$f(x)=x$，$cnt(x)=1$，否则$f(x)=1$，$cnt(x)=1$
- 枚举的质数$pr[j]$不能整除当前的i，那么显然i不是$pr[j]$的因数（废话），换而言之，$pr[j]$是$i*pr[j]$的最小质因子，那么此时$m=1$时$f(i*pr[j])=i*pr[j]$，$cnt(i*pr[j])=1$，否则$f(i*pr[j])=i$，$cnt(i*pr[j])=1$
- 枚举的质数$pr[j]$能整除当前的i，那么$pr[j]$也同时一定是i的最小质因子，因此$cnt(i*pr[j])=cnt(i)+1$

，同时当正好cnt出现m的倍数时$f(i*pr[j])=pr[j]*f(i)$，否则$f(i*pr[j])=f(i)$

至于最终数值大小，一定严格小于$\frac{(1+n)*n}{2}$，而n算到最大也就2×10^7，一定不会爆long long

总复杂度：$O(n)$

K. Kazusa's Party

Author: ZCQ

可以二分图匹配但没必要。

暴力枚举每个男生和谁匹配即可。

L. Little Witch Academia

Author: DHR

首先要发现这样一个事实：**对于某一层来说，合法的方案数与斐波那契数同阶**

由于墙的宽度很小，所以可以dfs找出合法的方案，你很快就能发现方案数最多也就 120 种

接下来用最多 $O(120^2\times w)$ 的时间可以处理出**任意两种方案能否放在相邻层**

我们维护一个矩阵 $ok[i][j]$，如果 i 方案和 j 方案能够相邻，那么我们标记 $ok[i][j]=1$，否则 $ok[i][j]=0$

同时也不难发现 $ok[i][j]=ok[j][i]$

现在仔细想想你获得了什么？这个 120×120 的矩阵可以被理解为一张巨大的无向图，总的砌墙合法方案数就是：**以任意两个点做为起点和终点，在无向图上经过 h 步之后的方案数之和**

这是矩阵快速幂的经典应用，总的时间复杂度为：$O(120^3\times log(10^9))$

这道题idea来自网易雷火去年的笔试题，原题只有长度 2 和 3 两种砖，并且宽度 ≤30，高度 ≤10，原题的第一步与本题差不多，但是无法用矩阵快速幂求解，因为合法方案数可能会大于 10^3，但是原题的高度非常小，所以可以在有向图上暴力dfs来统计答案

M. Minecraft

Author: DHR

题面中有两个重要信息：

- 沙子下落当且仅当它下面没有沙子或者它不在地面上
- 每次只能选择某一种字母全部填上沙子

考虑怎样的放置顺序才能让沙子不下落：

- 对于放在地面上的沙子来说，它们可以在任意时候被放置
- 对于没有放在地面上的沙子来说，它们必须在自己下面的沙子被放置好后才能被放置

我们不妨考虑用有向图来表示这种关系，低层沙子的标记向高层沙子的标记连边，对于那些入度为 0 的标记，我们显然可以在第一时间就放置它们

这很快让我们联想到**拓扑排序**

但是如果原图中有环是什么情况呢？标记 A 能放置当且仅当标记 B 已经放置，标记 B 能放置当且仅当标记 A 已经放置，显然这种情况是无解的

最后为了输出字典序最大，你需要在拓扑排序中使用**堆**来维护标记的入队出队

出题人觉得很绝望的一点是可能由于数据太弱，被很多回溯的dfs搞过去了

实际上本来是考虑要卡掉dfs的，相对来说拓扑排序要优雅的多

这道题定位是签到，idea来自于google去年kickstart的一道题。原题在二维平面上并且可以输出任意放置顺序，但是为了不写special judge加上了字典序一条规则，并且我发现三维和二维空间的解法没什么本质区别，所以改到了三维。不得不说google的出题水平太高了。