A. Maomao’s Candy

本题首先要求出 Dudu 最多能走的步数。

当 $\min(n,m)=1$ 时，Maomao 一定能抓住 Dudu；

当 $\min(n,m)>=2$ 时，若 $|r1-r2|+|c1-c2|$ 为奇数，则 Maomao 永远抓不住 Dudu；

若 $\min(n,m)=2$ 且 $|r1-r2|+|c1-c2|$ 为偶数，Maomao 只要一直往 Dudu 所在方向走即可抓住 Dudu；

若 $\min(n,m)>2$ 且 $|r1-r2|+|c1-c2|$ 为偶数，Maomao 只要将 Dudu 堵在地图四个角落之一即可抓住 Dudu。若两人当前位置 $|r1-r2|=|c1-c2|$ 时，Dudu 不可能跑到别的角落；当 $|r1-r2|$ 不等于 $|c1-c2|$ 时，Maomao 按照尽快使得 $|r1-r2|=|c1-c2|$ 且令 $|r1-r2|+|c1-c2|$ 不断减少的策略走，无论 Dudu 最终在哪个角落被抓住，Maomao 始终能保证以最少的步数接近该角落。而 Dudu 能做的是选择在哪一个角落被抓住。在 Dudu 能到达的角落中，Maomao 走到哪个角落所花步数最多，Dudu 最后将停留在哪个角落。Dudu 能不能走到某个角落视 Maomao 和 Dudu 的初始位置而定。

确定步数之后，Dudu 在每一步中的吃的糖果满足斐波那契数列。我们可以发现，Dudu 所吃糖果的总数也满足类似斐波那契数列的性质。Dudu 前 n 步所吃糖果总数 $F(n)$ 满足：

$F(n)=G(n)-1$；

$G(1)=2$；$G(2)=3$；$G(n)=G(n-1)+G(n-2)$（当 $n>=3$ 时）

矩阵快速幂求出即可。由于取模数字过大，中间过程要用到慢速乘。

B. Dudu's maze

由于只有一个 magic portal，先把当前所在的连通块的糖果全拿走，之后在第一次遇到敌人的时候使用肯定是最优的。

先用并查集将没有敌人的点合并起来，用一个数组存放该连通块所含点的数量（称作权值），再用搜索搜出和起点所在连通块相连的怪物点，把这些怪物点枚举一下，遍历连接这个怪物点的所有边，求边的另一端的连通块的权值（1 这个点所在连通块的权值要改为 0，因为已经拿过了），求和之后再除以与该点相连的边数，取最大值，加上 1 所在连通块的权重即可。

因为遇到敌人是随机选择边逃跑，因此遇到重边的时候连通块的权值要累计多次。同时也是除以连接的边数而不是连接的连通块数量或者房间数量。拿到的糖果数量至少是 1 所在连通块的权重值，不论经过哪里，都一定能拿到那些糖果，所以最后答案直接加上即可。

C. Dawn-K's water

完全背包加枚举,设 f[i]表示恰好买 i 单位的水最少花费的价格(如果无法恰好买 i 单位则赋值无穷大)
因为题目要求的是至少买 m 单位水的质量所花的最大价格,并且答案价格小于 1e4,所以在用完全背包求出所有的 f[x](1<=x<=1e4)后
从 m 开始往后更新答案即可: 若 ans>=f[i], 则 ans=f[i](m<=i<=1e4).
由于 n 的总和<=5e4,答案又保证了<=1e4,所以时间复杂度大概为 O(5e8).

D. Fish eating fruit

题意：树上任意两点之间的路径按照模 3 为 012 分类，将两点间距离加和，乘 2 即为答案。

根据模 3 的余数设计 dp 方程
dp[i][k]统计距 i 模 3 为 k 的子节点的数目
fp[i][k]统计距 i 模 3 为 k 的非子节点的数目（父节点，兄弟节点，兄弟节点的子节点）
ans[i][k]统计距 i 模 3 为 k 的子节点到 i 的距离和
fans[i][k]统计距 i 模 3 为 k 的非子节点距离和

设计一下转移关系就可以过题了。

```
dp[root][(j+z)%3]=(dp[root][(j+z)%3]+dp[k][j])%MOD;
ans[root][(j+z)%3]=(ans[root][(j+z)%3]+(ans[k][j]+(dp[k][j]*e[root][i].dist)%MOD)%MOD)%MOD;
fp[k][(z+j)%3]=(dp[root][j]-dp[k][(j-z+3)%3]+fp[root][j]+MOD)%MOD;
fans[k][(z+j)%3]=(fans[k][(z+j)%3+(ans[root][j]-ans[k][(j-z+3)%3]-dp[k][(j-z+3)%3]*e[root][i].dist+3*MOD)%MOD+fans[root][j]+fp[k][(z+j)%3]*e[root][i].dist)%MOD;
```

答案就是所有节点的 fans+ans 和。

E. Gugugu's upgrade schemes

首先容易看出这个数是集合的划分数，即贝尔数，对于贝尔数的预处理一般需要 O（n^2）的复杂度，显然对于题目里的数据范围是不合适的，但是我们观察到模数是一个小于 1000 的质数，以此为突破口，如果知道 Touchard 同余的话这题就迎刃而解了，Touchard 同余简而言之就是当 p 为质数时 ，这样我们可以预

处理出小于 p 的贝尔数，然后可以通过记忆化的方式去搜索，因为考虑到递归过深可能爆栈，故此题限制 n/p<1000.

PS：出题人再次在这里各位道（xie）歉（zui）了，因为数据的原因，导致部分队伍 TLE 了，并在这题上浪费了太多时间，这里感谢 Clarifications 里有朋友能及时指出问题，让我们能及时修正错误，但给大家带来不好的参赛体验，真的是非常抱歉，不过还是欢迎大家能来参加 ICPC 沈阳站的，希望大家能在现场赛上取得好成绩!

F. Honk's pool

题意：每次挑出最大的，令其减一，然后挑出最小的，令其加一。共操作 K 次，求最大值和最小值的差。

考虑到 k 的范围，不能进行模拟，尝试去二分。

先求一下平均值，因为随着取水的次数，所有池塘的水量是趋于相同的。也就是超过平均值的池塘的水会变少，低于平均值的池塘的水会变多。

分别二分最后状态的水最多的池塘的水量和水最少的池塘的水量。

我们二分到最后求出来的值就是在 K 次操作内最大值的最小值和最小值的最大值.

也就是在 K 步操作(以及之内)不可能将最大值和最小值的差进一步缩小,那么此时最大值和最小值的差就一定是最优的答案.

不过本题需要考虑好二分的边界.假设 K 足够大,那么最后一定能达到平均.

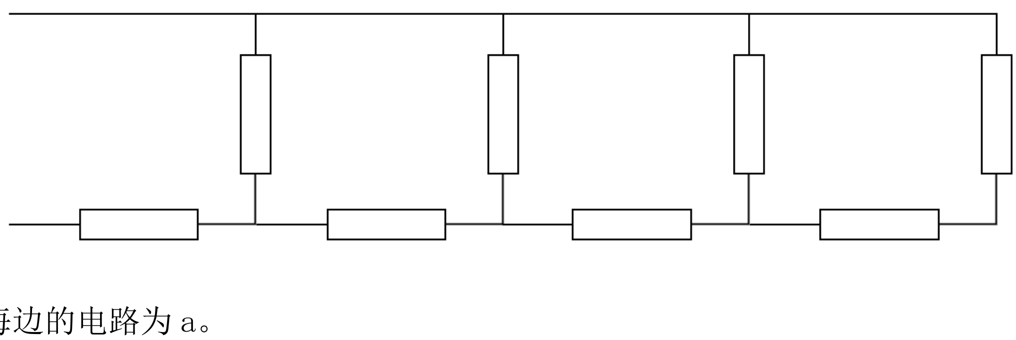
假设所有 ai 的和为 sum，如果 sum%n==0,则所有水池的水会一样多,此时最大值的下界和最小值的上界都是 sum/n。

而如果不能整除，则最小值的上界不变，最大值的下界是 sum/n+1。

G.　Special necklace

本题有一定的物理背景。首先注意到红宝石的电阻是实心的，而任意两端点电阻阻值相等。所以可以构造电阻为三条完全相等的电阻首尾相连为一个三角形。（可以由电路原理验证）

之后可以由电路的*等效变换将电路化为:*



每边的电路为 a。

之后可以由电流的递推关系得到一个电阻通项公式：
R=(sqrt(5)/(pow(cur,n)-1)+(1+sqrt(5))/2.0)*a
其中 cur=(3+sqrt(5))/(3-sqrt(5))
但是 n 非常的大，以至于任何方式都处理不了。算出当 n 趋于正无限时，用“加一个节等于原来的电阻”的方法可以得出：R=(1+sqrt(5))/(2*1.0)*a.
而当 n 大于等于 10 时，通项结果与 n 等于正无限时的电阻差值小于 1e-6。
得到这个结论之后，要么直接按照该公式书写，要么大胆的采用暴力模拟的方法就可以过题了。

H. Texas hold'em Poker

虽然题目描述复杂，但实际上只是情况较多，直接模拟即可，考虑到牌的数量只有 5 张并且各种情况也有限可以使用桶排序。

I. Self-game

由于所有局面的总数不是很大，我们可以把每一个局面看成点
对于一个局面因为有 ghh 行动和 gfh 行动两种不同的情况，可以把一个局面拆成两个点，
于是形成了一个二分图，对于 i 和 j 两个可以互相变换到的局面，在二分图中对称地连边。
如何知道一个局面是否为先手必胜呢？
其实求一个最大匹配就知道了，然后对 ghh 的先手局面 i 进行讨论：
如果 i 没有和任何点匹配，那么必败。（gfh 不断沿着实边匹配边走）
如果 i 与 j 匹配，当且仅当存在一条由 i 出发的交错路，满足在交换路上的实边与虚边之后，匹配数不变，这时才必败，否则必胜。（gfh 不断沿着虚边走）
具体实现可参看 std
注：也可以将一个局面两颗棋子的坐标（x1,y1,x2,y2）按照和的奇偶性建立二分图，用类似的方法讨论，这样常数更小，不过鉴于复杂度是一个级别的，所以 std 就没有进行这样的改进了。
时间复杂度：O（n*m+k*m）

J. Ghh Matin

由于每个点的入度都等于出度,则这个图一定是一个或多个欧拉回路组成,要求从任意一个点出发最后都能回到自己,当且仅当所有回路的长度不超过 x.
比起计算所有回路都不超过 x,计算有一个回路超过 x 更容易,因为保证 x>=n/2,

所以有一个回路长度为 x 和有一个长度为 x+1 之类的事件是互斥的,所以总概率是他们的和.
设 m 为回路长度,当 m>x 时， 只需要先选出 m 个点构成环,剩下的元素再随机打乱就行
所以 P(m)=c(m,n)*(m-1)!*(n-m)!/(n!)=1/m
所以所有长度不超过 x 的概率为 1-∑p(i) (x<i<=n)

K. Guanguan's Happy water

首先 f[i]=a[i] (1<=i<=k)
然后可以确定递推公式 f[x]=f[x-1]*p[1]+f[x-2]*p[2]………+f[x-k]*p[k];（x>=k）

由前 2k 项可以构造方程组：
f[k+1]=f[k]*p[1]+f[k-1]*p[2]……+f[1]*p[k]
f[k+2]=f[k+1]*p[1]+f[k]*p[2]………+f[2]*p[k]
………
f[2*k]=f[2*k-1]*p[1]+f[2*k-2]*p[2]……+f[k]*p[k]

然后高斯消元求出这个 p 序列
根据之前求出的 p 的序列生成系数的矩阵 A。
可以得到：

$$A^{n-k} * \begin{matrix} f[k] \\ \dots \\ f[1] \end{matrix} = \begin{matrix} f[n] \\ \dots \\ f[n-k+1] \end{matrix}$$

将 n-k 个式子全列出来相加,利用结合律会得到：

$$A^1 * \begin{matrix} f[k] \\ \dots \\ f[1] \end{matrix} + A^2 * \begin{matrix} f[k] \\ \dots \\ f[1] \end{matrix} + \dots + A^{n-k} * \begin{matrix} f[k] \\ \dots \\ f[1] \end{matrix} = (A^1 + A^2 + \dots + A^{n-k}) * \begin{matrix} f[k] \\ \dots \\ f[1] \end{matrix} = \begin{matrix} f[k+1] + f[k+1] \dots\dots..+f[n] \\ \dots\dots \\ f[2] + f[3] + \dots..+f[n-k+1] \end{matrix}$$

因此只要求出$A^1 + A^2 + \dots + A^{n-k}$就能得到 f 的前 n 项和。（就是求等比矩阵的前 n-k 项和）

构造分块矩阵：$S = \begin{matrix} A & E \\ 0 & E \end{matrix}$

对 S 取幂可得：

$$S^2 = \begin{matrix} A^2 & A+E \\ 0 & E \end{matrix}, \quad S^3 = \begin{matrix} A^3 & A^2+A+E \\ 0 & E \end{matrix}, \dots\dots S^x = \begin{matrix} A^x & A^{x-1} + \dots + A + E \\ 0 & E \end{matrix}$$

用矩阵快速幂算出S^{n-k+1}，右上角的部分减去 E，就是$A^1 + A^2 + \dots + A^{n-k}$， 然后根据前面推的关系得到答案。复杂度 $O((2*k)^3 * \log(n)$