A String

一种做法可以直接用$exkmp$求出S串和它的所有后缀的LCP,那么我们令$[1,n]$和$[i,n]$这段LCP长度为x,当满足$i \leq x$时都会在$(i-1) \times 2+k, (i-1) \times 2+2k \ldots ((i-1) \times 2+xk \leq i+x-1)$产生贡献,因此做一个模意义的差分即可。

本题也可以用kmp树做链上的模意义差分,对于一个前缀i,我们需要找到所有kmp树上的祖先x并满足$(2x-i)\%k==0$,这个可以直接在kmp树上维护出来，就是从当前点向上的一条链，可以用kmp求出每条链的链顶(方法就是kmp的时候保证$border$长度不超过串长的一般)，然后再dfs一遍求解答案。不过由于评测系统的栈空间有限，可能一些dfs的写法出现了爆栈的情况。

B Dragon slayer

签到题。墙的数量只有15，爆搜或者状压都能比较轻松的通过。

C Backpack

$f_{i,j,k}$表示前i个物品，异或和为j，并且体积为k的方案是否存在。

$f_{i,j,k} = f_{i-1,j,k} \vee f_{i-1,j \bigoplus v_i,k-w_i}$

$f_{i,j}$实际上是$f_{i-1,j}$和$fi-1,j \bigoplus v_i$左移w_i位后$\vee$，可以用bitset压位。

D Ball

显然我们需要判断出一个数是不是质数，由于范围只有$2*10^5$,各类筛都可以

接下来考虑枚举一下三个点，我们可以分别计算距离并进行判断

但这样子时间复杂度太高，考虑枚举两个点，如何确定第三个点，我们即要求其中一条边小于等于这条边，另一条边大于等于这条边，这可以利用两个$bitset$的$\&$解决

处理最终答案的时候可以把边从小到大加入避免算重。

E Grammar

题目大意就是有n个特殊符号，26个普通符号（小写字母），特殊符号可以推导出一个包含特殊符号和普通符号的符号串，每次询问从第x个特殊符号开始推导出得符号串中第y个字母。

首先我们对每个特殊符号建一个点，对于推导出来的情况，类似1->234可以连2->1 3->1 4->1这样的边，对于这张图我们可以用拓扑排序算出每个式子的长度。

当然这其中有些式子的长度是无穷的，那么我们需要找到这个产生式中第一个能够让其产生循环的位置(特殊符号)，并向这个特殊符号连边。

最终会产生一棵内向环套树，不过这不重要，对于一个询问，我们从第x个符号串开始，二分出第y个位置在哪里或者第y个位置在哪个符号串里，然后暴力往下做就可以了，这样做下去每个串最多访问一次，复杂度$O(n^2logn)$，实际跑的很快。

由于n的范围比较小，直接记忆化搜索也是可以过的。

F Travel plan

由于不存在长度为偶数的简单回路，所以任何两个简单回路不存在公共边。

一个不严谨的证明：

如果某两点之间存在三条不重合的简单路径，则三条路径中必然存在相同奇偶性的路径，就会存在偶数长度的简单回路。

不存在偶数长度的简单回路，说明必然任意两点之间都不存在三条不重合的路径，即任意一条边只会存在于一个简单回路上。

于是这个图就是仙人掌图。

求最大公因数为x的路径，可以用莫比乌斯反演转化为求x的倍数的路径。

这样对只需要对n个只包含i的倍数的边的图计算简单路径数量。

仙人掌图上的简单路径数量可以用dp求得。

G Treasure

首先建出并查集重构树(在最小生成树的基础上，每次新增一条边就加一个新点连向合并的两个集合)，对于一个询问，我们利用倍增跳到对应的节点上，那么问题就变成了查询一个子树。

那么对于这一棵子树，我们需要查询子树里每种类型的最大值并求和。

因为每种类型的点最多只有10个，我们在重构树的每个点维护子树每种类型最大值的和，那么可见对于一次修改，实际只需要做不超过10次链上加操作，因此只需要像建立一个类似虚树的结构，每次询问从修改点暴力往上跳并且做链上修改即可，这个过程可以用树状数组维护(链上加单点求值在树上差分后变成单点修改子树求和)，总时间复杂度$O(10nlogn)$。

H Path

对于题目英文表述不清楚出题人深感抱歉

首先对于上一步是不是特殊边我们可以考虑建一个分层图，上面一层代表上一步是，下面一层代表不是

注意到如果没有0的边这个题就是普通的最短路

考虑走到上一层之后我们应该怎么走0的边,我们可以把没访问过的点加入set,并取出所有点并判断这些点是否存在于当前点出边集合，如果不存在就将其最短路更新，更新成功则入堆

比较显然可以证明复杂度是$O((n+m)log)$

I Laser

首先如果所有点都在同一条水平/竖直/斜线上，那么可以直接输出。

假设我们已经确定一个点在水平方向上(米字的一横)，那么随便找一个不在这条直线上的点，用竖线和两条斜线可以交在这条线的三个点上，只需要暴力判断这三个点作为中心点是否合法即可。

对于更普遍的情况，我们只需要每次将坐标旋转45度，将上面的过程重复4次即可，这样可以讨论到所有情况。

或者可以直接找到两个不共线的点，讨论两条线的相交情况，做12次判定也是可以的。

时间复杂度$O(nlogn)$。

J Walk

首先计算可发现$f(y)=S(S(S(y)))\leq 2$

我们考虑容斥，去计算$(x+1,P)$ 满足$p>y+f(y)$

这样子我们会发现 我们走的路径一定满足是向上单调递增的

记$f(x)$表示满足容斥条件的函数,$g(x)$表示最终答案

则$g(x)=\sum_{i=0}^{n-1}(-1)^{n-i-1}f_{n-i}*g_i$

令$h(x)=\sum_{i=0}^{n}(-1)^i f_i x^i$

可以得到$g(x)=\frac{1}{1-h(x)}$

现在我们考虑怎么求$f(x)$

由于数值具有单调性,所以我们可以考虑对于数值进行分治计算

对于一段$[l,r]$,记$a[i][j](1\le i,j\le 3)$ i表示左边最近选的点距离左端点的距离,j同理

于是这样子我们可以利用分治ntt合并

注意对于长度$(len\le 6)$的区间需要单独dp出答案

实现有一些细节，具体可以看std

K Random

答案为$(n-m)/2$

求出2的逆元即可

L Alice and Bob

将每个值为i的数字视为2^{n-i},那么Alice的胜利条件就是最终局面中能出现2^n。

Alice将数字分成两个集合，Bob将其中一个集合减一，就相当于将这个集合中的数全部乘2，然后将另一个集合删去。

如果Alice能将集合中的数字按照值二等分，那么无论Bob怎么操作，黑板上所有数字的总和实际是不变的。

如果集合中的数字总和超过2^n,由于所有数字都是不超过2^n的2的幂次，那么Alice的每次分割总能使得两边集合的值均不小于2^{n-1}。

因此直接判断所有数字的2的幂次的总和即可。