这道题在本质上是一道暴力搜索题。

①找出一个子集，使其所有数的按位xor为零。

②由于原先所有数的按位xor为零，这意味着剩下那些数的按位xor也是零。

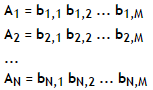
③我们希望让选出的子集尽量小。

④我们希望最优化N/K。

现在问题变成了：能否有效率地选出一个按位xor为零的子集？

当然可以。关键就在于F2域（即01）上的高斯消元法！

假如我们把N个数的二进制表示写出：



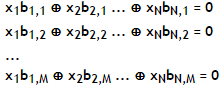
其中每个数至多含M个二进制位。

定义一个每一位是01的N维未知向量：



对于最初N个数的某一子集S，我们令xi=1当且仅当Ai∈S。

现在，由于集合中所有数的按位异或值必须为零，如下方程组必然成立：



其中X表示了S中有哪些数。

我们已知对于X={1：重复N次}，上述方程组成立。

我们有N个未知数，但只有M个方程。（有时N可能和M相等或甚至更小，因此小心！）

那么，X可能有多个解。

①如果一个解中1的个数比较少，我们就称它比较小。

②我们希望找出一些互不相交，而且比较小的解。

③我们可以找出一个小解。

④然后忽略掉这些xi。

⑤然后重复③~④。

如此一来，我们就可以将原先的N个数拆分。

现在问题在于，如何找出一个小解？

答案是：

proc: 找出答案

1: 选择某个 i, 使得 x[i] 尚未被选

2: 我们假设 i 必须在选出子集中，并在方程中固定相应的的二进制位

3: 从未选x[i]中随机取出 O(M) 个变量

它们组成含M个方程M个未知数的方程组，检查该方程组是否有解

若有解

标记这些x[i]已被选

若无解

重复步骤3

我们可以科学地挑选那O(M)个变量来优化这一方法。

上述程序中需要用到F2上的高斯消元法。这意味着：

①加减操作只是简单的异或

②乘法仍然是乘法

③由于所有值要么是0要么是1，我们不需要做除法

④如果没有系数1，或者答案应该是1，那么方程就是退化的。

⑤方程可能线性相关，但这并不重要，因为我们仍可忽略掉无关变量来得到一组解。