# 鸽子

清华大学 E.Space

# 鸽子和鹦鹉的故事

- 本题确实是试机题IOI2011鹦鹉的改编
- 把打乱顺序改成只有比较近的打乱顺序就是这道题了
- 想到OIer中最经常提到的鸟类是鸽子，就拿鸽子做了题目名称
- 这大概是NOI系列比赛中的第一道通信题（?
- 在IOI和JOI中都出现过通信题
- 由于相关内容是NOI大纲10级，就只能放CTS了

# 题目大意

- 你需要传输一个128位的01串
- 信道的干扰是发送的01串中每一位的顺序可能偏移不超过 $k$ 位
- 求一个尽可能短的编码解码方式

# 子任务 0

- 样例
- 提交样例代码即可得分
- 用于熟悉本题的任务是什么

# 子任务 1

- $msg < 1024$
- 注意交换不改变0和1的个数
- 当鹦鹉做就行
- 是几就用几个0

# 子任务 2

- $msg < 1048576, k = 1$
- 一种想法是考虑到不会移太远，所以1的位置和不会改变太多
- 对于每个 $i$ 算一下有 $i$ 个1的时候最小位置和和最大位置和，再除以 $3i$，发现就比 $1048576$ 大了
- 当然可能有一些其它差别多能做这个范围的做法

# 评分参数的由来

- $C(k)$ 的值就是标算用到的长度
- 由于可以按照长度给分，就直接这么定了
- 当然欢迎吊打标算
- 这条分数曲线大概是综合了几个不同的做法，按照我认为的这几个做法应得的分数定的

# 算法 1

- 每个数活动的范围只有 $2k + 1$
- 可以用 $2k + 1$ 个数表示一位
- 周围全0，中间一位表示真正的内容
- $n = 128(2k + 1)$
- 可以去掉头尾减少一点长度
- $n = (384,640,1408,1920,2688,3712,5248)$
- 得分在 25 左右

# 算法 2

- 考虑第一位填什么
- 那么只要后面 $k$ 位和第一位一样，第一位就不会改变
- 于是可以考虑把每一位重复 $k+1$ 遍
- 可以去掉最后 $k$ 位
- $n=(256,384,768,1024,1408,1920,2688)$
- 得分在 43 左右

# 算法 3

- 还是可以考虑0和1个数的问题
- 注意到如果出现连续 $2k$ 个0或1，那么从这 $2k$ 个数正中间切开，前后两段0和1的个数不会变
- 考虑这样构造：
- 每一段为前 $k$ 个数一样，后 $k$ 个数一样，中间插入 $k-1$ 个数用来调整这一段中0和1的个数
- 需要保证相邻两段中前一段的最后一个字符等于后一段的第一个字符

# 算法 3

- 假设第一段以 $k$ 个0开始
- 这样所有段的前 $k$ 位就确定了
- 若是0，考虑数整段中1的个数
- 若是1，考虑数整段中0的个数
- $k-1$ 个任意字符和 $k$ 个连续相同字符的组合可以组成一共 $2k$ 种不同的数量
- 所以可以用 $3k-1$ 位表示 $2k$ 的范围
- $k=1$ 的时候可以多插一位，这样编码更短

# 算法 3

- 需要做一下进制转换
- 可以去掉最前面的 $k$ 位和最后若干位
- $n = (242,318,535,665,847,1074,1417)$
- 得分在 62 左右
- 其实这个是本题一开始的标算

# 算法 4 by zjc

- 其实这个做法比刚才的算法 3 还好想
- 注意到存在 $2k+1$ 个连续相同的数的时候，中间一个的位置一定是不变的
- 考虑把1和0看成两个不同的角色
- 让0可以在任意位置出现，但1只能连续 $2k+1$ 个一起出现
- 解码的时候只要找到每 $2k+1$ 个1中的第 $k+1$ 个就行
- 这样长度为 $n$ 的编码能够表示的消息的数量是一个线性递推
- $f(n)=f(n-1)+f(n-2k-1)$

# 算法 4 by zjc

- 数字和序列之间的转换可以简单写个DP
- $n = (234,318,531,656,829,1045,1348)$
- 得分在 68 左右

# 算法 5

- 算法 4 显然不是完美的
- 编码中1的个数必须是 $2k+1$ 的倍数
- 这就意味着大量可用的编码都没有用到
- 考虑这个题的本质是什么
- 独立集问题!
- ~~“小 E 喜欢出最大权独立集问题。” （出处：CTS/省选2022）~~
- 本质相当于选择一个编码的集合，使得这个集合的任意一个编码在两步的变换之内不能变成另一个编码（类似于海明码等问题）

# 算法 5

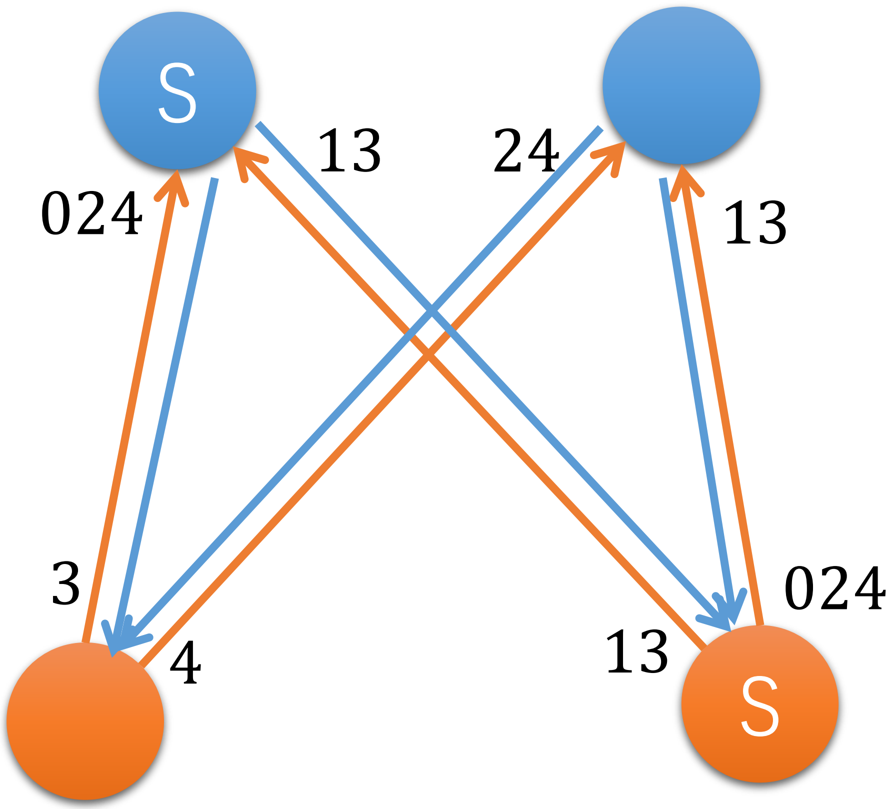
- 我们保留算法 4 中的所有编码，考虑还能再加一些什么
- 我们还是希望多限制1，少限制0
- （直觉上线性递推中 $f(n-1)$ 一项对特征根的影响很大）
- 考虑连续相同数字的长度会对编码的选取产生什么影响
- 一段长度在 $[1,2k]$ 之间的1可以在两步之内和一段长度在 $[1,k]$ 之间的0交换位置
- 这也说明了算法 4 中为什么1的长度选择用 $2k+1$

# 算法 5

- 11111|110000|00111|11111
- 11111|000011|11100|11111
- 我们可以发现，中间这段0的长度需要至少是 $3k+1$
- 所以当一段1的长度不是 $2k+1$ 的倍数的时候，后面的0长度至少要是 $3k+1$
- 同时当一段0或1的长度不超过 $2k$ 时，前面一段的长度需要是 $2k+1$ 的倍数
- 这样可以构造出如下的状态机：

# 算法 5

- Type 0: $[1,2k]$
- Type 1: $2k+1$
- Type 2: $[2k+2,3k]$
- Type 3: $\{m(2k+1) \mid m \in N, m \geq 2\}$
- Type 4: Others



# 致谢


- 感谢左骏驰提供了算法 4 让我加强了标算。
- 感谢王逸松提供了能够测试通信题的评测环境。
- 感谢彭思进对题面的一些建议。

# Q&A