数卡模型探究

及程序设计

摘要：本文结合数卡模型的特点，立足题中“单面”和“正反双面”两种数卡的不同要求，分别将十进制数字转换为二进制和三进制数字，提出确定数卡模型的编制方法，并针对“双面”数卡在不同识别量下的容量的合理取值通过局部调整法，建立了动态优化模型，逐步优化，给出了输入识别量便可生成下界以及与最小容量对应的数卡集的程序。

吴昊	数学系	131110017
刘铭	商学院	131090180
陈云龙	软件学院	131250181

2015高教社杯全国大学生数学建模竞赛
承 诺 书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称为“竞赛章程和参赛规则”，可从全国大学生数学建模竞赛网站下载）。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛章程和参赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从A/B/C/D中选择一项填写）： B

我们的报名参赛队号为（8位数字组成的编号）：

所属学校（请填写完整的全名）： 南京大学

参赛队员（打印并签名）：1. 吴昊

2. 刘铭

3. 陈云龙

指导教师或指导教师组负责人 （打印并签名）：

日期： 2015 年 7 月 20 日

赛区评阅编号（由赛区组委会评阅前进行编号）：

2015高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅编号（由赛区组委会评阅前进行编号）：

赛区评阅记录（可供赛区评阅时使用）：

评阅人										
评分										
备注										

全国统一编号（由赛区组委会送交全国前编号）：

全国评阅编号（由全国组委会评阅前进行编号）：

数卡问题探究

摘 要

本文结合数卡模型的特点，立足题中“单面”和“正反双面”两种数卡的不同要求，分别将十进制数字转换为二进制和三进制数字，提出确定数卡模型的编制方法，并针对“双面”数卡在不同识别量下的容量的合理取值通过局部调整法，建立了动态优化模型，逐步优化。

对于问题 1，我们用二进制 1，0 分别表示数字在或不在单面印刷的数卡的两种状态。用二进制表示正整数，将所有对应位数为 1 的数字放入同一张卡片，升序排列。类似，在问题 2 中，我们用三进制 1，2，0 表示某个数字在双面卡片上的三种状态，方法是用三进制表示正整数，将所有对应位数为 1 的数字放入同一张卡片的正面，对应位数为 2 的数字放入同一张卡片的反面，升序排列。

由此我们可以确定两种数卡在给定识别量（问题重述中有定义）条件下的最小卡片张数，也给出了一种依据进制转换的编制方法。对于识别量为 80 的双面数卡，我们运用集合、映射的思想，通过数学归纳法证明了这套数卡的唯一性。

对于识别量分别为 100 和 160 的双面数卡，我们无法通过全部遍历的方法，找到最小的容量值。因而，我们采用了动态规划模型，使用了局部调整法，通过循环算法、对出现重复数字从容量上逐步优化，最终得到了容量值分别为 25 和 48 的较优解。而同时，我们从数学证明了识别量为 100 和 160 的双面数卡的下界为 25 和 46，由此我们可以发现，对于容量为 25 为识别量为 100 的最优解。

问题 4 尝试用数卡模型解决实际问题，对此，我们分别从二进制和三进制角度，对数卡模型在二维码和密码学领域的实际应用进行了简要的探索。

附加题则是对前三题的进一步探索和研究。首先我们把对 n 计算卡片的个数 k 和理论下界 r 的方法延拓到整个正整数上。此外，我们还通过对离散的 n 的观察和计算得到了对一个区间上的 n 都适用的数组控制范围，从而把通过程序计算具体的数卡解答，并且得到目前优化方案下的最小值以及对应的数卡的方法延拓到 1-242（也就是 3^5-1）

关键词：进制；动态规划模型；局部调整法；数学归纳法；映射

1. 问题重述

对于一种单面数卡，我们约定同一面上的数按递增顺序排列。数卡中存在的任意一个数字的值，都等于包含该数字的所有卡片的最小值之和。同样，对于正反双面的数卡，我们约定同一张数卡两面不能有相同的数字，数卡上任意数字的值仍等于包含该数字的所有卡片的最小值之和。

为了叙述方便，我们给出关于“识别量”的定义。如果一套数卡满足下述条件：

（1） 数卡中出现的数的集合是$\{1,2,3,......,n\}$；

（2） 如果是双面印刷，同一张数卡两面不能有相同的数；

（3） 对于任意整数$1 \le x \le n$，所有包含x的那一面中最小的数（即左上角的数）之和恰等于x；

（4） 满足上述条件的数卡的张数k最少，那么我们称这套数卡的识别量为n。所有数卡中每一面数的个数的最大值称为数卡的单面容量，简称容量，用r表示。

对于双面数卡，我们可以进行这样的优化改进，在满足数卡张数k最小条件下，适当增加识别量n，且保证数卡容量r尽量小，数卡单面容量较为均衡。根据单面和双面数卡的编制原理和方法，我们要解决下列几个问题。

1. 用二进制原理解释单面数卡编制方法，计算数卡张数为k时，识别量n的最大值并证明。
2. 对识别量$n=80$的双面数卡，计算最小卡片张数k，说明不计顺序的这样一套数卡的唯一性并给出证明。
3. 对识别量$n=100$和$n=160$，编制出满足条件的，保证容量r尽量小的双面数卡。
4. 分析讨论上述结果在实际问题中的应用。
5. 对识别量为正整数n的双面数卡，试给出容量r的最小值m或它的下界，并说明数卡的编制的方法或在附录中给出计算机程序。

2. 问题分析

2.1 对于问题 1、2 的分析

2.1.1. 基于二进制和三进制原理的编制算法

数卡模型能够对识别量以内的正整数的数值，通过计算包含该正整数的所有卡片的最小值的和得到。对于单面数卡，该正整数“是”、“否”在卡片上，为两个状态。而对于双面数卡，该正整数“在正面”、“在反面”、“不在卡片上”为三个状态，于是可以考虑把数卡问题抽象为进制问题。据此，我们提出一种基于二进制原理的单面数卡的编制算法，并将其推广到基于三进制原理的双面数卡的编制算法。

- 单面数卡编制算法

1) 将十进制正整数转换为二进制表示形式，如 15 可以表示为 1111；
2) 把卡片上的最小值表示为二进制 1,10,100,1000,10000...， 如果某个数在二进制表示下某些位数是 1，那么它就会出现在对应的数卡上。此时这些数卡上的首位数字相加则恰好就是这个数。即，将对应位数为 1 的所有整数放在同一张卡片上；
3) 由此我们也可以看出，对于识别量为 n 的单面数卡，最小卡片数 k 值满足 $2^{k-1}-1<n\le 2^k-1$。

- 双面数卡编制算法

1) 将十进制正整数转换为三进制表示形式，如 15 可以表示为 120；
2) 把卡片上的正反面的最小值表示为三进制 1，2；10，20；100，200...(其中认为带有 1 的三进制表示为卡片的正面，带有 2 的三进制表示为卡片的反面)，同样，如果某个数在三进制表示下某些位数是 1，那么它就会出现在对应的数卡的正面。如果某个数在三进制表示下某些位数是 2，那么它就会出现在对应的数卡的反面。即，将对应位数为 1 的所有整数放在同一张卡片的正面，对应位数为 2 的所有整数放在同一张卡片的反面；
3) 同样我们也可以得到类似的结论，对于识别量为 n 的双面数卡，最小卡片数 k 值满足 $3^{k-1}-1<n\le 3^k-1$。

2.1.2. 基于识别量为 3^k-1 的双面数卡唯一性证明

对于问题 2 中识别量为 80 的双面数卡，$80=3^4-1$，即 4 张卡片刚好可以表示出 80 个正整数。由于一种选法只对一个数字，因此选法和数字之间的映射是一个双射。对此，我们结合集合、映射等相关知识，运用数学归纳法给出排列方法唯一性的证明。

2.2 对于问题 3 的分析

对于前两个问题，我们都可以给出具体的数学证明过程，但对于第三个问题，我们无法通过具体的数学推导过程找到最优的容量r值。此外，试想如果我们采取遍历法，比如当$n=100$时，可以确定最小卡片数目$k=5$，假定这5张卡片正反的最小值已给出，根据题意，一共可以组合得到3^5=243个数字，除去超过100的不符合题意的数字后，也很有可能得到几十组重复数字，不妨设重复的结果数为n，满足$n\geq 20$。这意味着即使我们给定了十个数字，也有可能生成2^n（$n\geq 20$）组数据。如果考虑5张卡片上正反的最小值的取值情况，问题将变得更加复杂，数据量会远远超出我们可通过编程运算的范围内，无法通过遍历的方法找到最优解。

因此我们需要进一步优化算法，找到较优的容量r值。在解决这个问题上，我们主要运用了动态规划的思想和方法，结合局部调整法逐步逼近较优容量r值。首先，我们可以通过某种方法给出初值和5张卡片正反较好的一组最小值的取值范围。同时，我们采用多线程的编译方法保证运算速度，在每次遇到前面所提到的重复的数值时，优选含有重复数值的两组数据得到的容量r的较小值，次选两组数据得到的每张卡片容量的和，以及选取这两种数据最终生成的卡片容量的方差较小值，舍弃重复数值的另一个。此外，我们还进行多步循环，多次优化。最终我们得到了在识别量$n=100$时的较优容量$r=25$和$n=160$时的较优容量$r=48$。

而通过数学证明，我们分别得到识别量$n=100$的下界25和$n=160$的下界46。

由此可以看出，$n=100$时，最优容量$r=25$，并且我们可以给出多组满足条件的双面数卡；而$n=160$时，我们能得到较优容量为$r=48$的双面数卡。

2.3 对附加题的分析

在第三个问题中，我们已经能够在使用五张卡片的情况下，对于特定的两个数值，通过动态规划的调整法得到较优容量r值。我们发现这样的做法具有普适性。因此我们再次在某些特定的点（指的是识别量为81，90，110，120，140，160，190，210，242）上利用程序进行试验，并加以计算，最终得出对多个不同的连续的n都适用的数组范围。

最终我们把五张卡片的情形分成了三类，每一类各使用一种范围。并把这样的做法很容易推广到了卡片数更少的情况。

在需要一张或者两张卡片时，因为情况简单，可以直接给出结果。在需要三

张或者四张卡片时，由于情况不多，所以都只采用了一个公共范围。

3、模型的建立和求解

一、基本符号说明

n：识别量

k：数卡的张数

$r_1,r_2,\cdots r_{2k}$：双面数卡上 $2k$ 个面上的数字个数

r：$\max\{r_1,r_2,\cdots r_{2k}\}$

$x_1,x_2,\cdots x_{2k}$：双面数卡上 $2k$ 个面上的每一面的最小数

二、模型的假设

这个问题的核心在于第三问（延拓到附加题）。第一问和第二问中我们只要运用到二进制和三进制的方法就可以得到结果。而在第三问中，我们需要我们把它分为两步。第一步是得到比较好的数组$(x_1,x_2,\cdots x_{10})$的范围。第二步是对固定的$(x_1,x_2,\cdots x_{10})$得到尽量优化的解。其中第一步是通过数学计算对$x_1,x_2,\cdots x_{2k}$给出一些限制条件，方便第二布的进行。而第二步则是一个动态规划问题。我们把一个决策数问题（不同加法方式的选择）变换为一个最优化问题（逐步调整）。在这个选择中，每个和的不同加法方式选择方法是决策变量，而最终得到的r的最小值是状态变量。我们采用局部调整的办法。把每一个决策变量反复一一调整，而调整方法是考察关于$r_1,r_2,\cdots r_{2k}$的三个函数，依次根据最大值、求和、方差进行比较得到最应取的。最终得到可以得到的r最小的解。

三、问题的解答和模型的建立

3.1　对于问题 1 的解答

3.1.1 用二进制原理解释数卡的编制方法

我们把 1 到 60 这 60 个数按二进制写出，分别是 1，10，11，100，101，110……111100.把这 60 个数字根据二进制的表示形式写在数卡上。先看末尾，如果是 1 就写到第一张数卡上，如果是 0 就不写。再看从右往左第二位，如果是 1 就写到第二张数卡上，如果是 0 就不写。第三张数卡则看从右往左第三位。根据这样的方法依次把这些数字写在六张数卡上。

那么第一张数卡，则是所有的奇数。即，1，3，5，7，9……

第二张数卡，则是所有除以 4 余 2 或 3 的数。即，2，3，6，7，10……

依次得到如题所给的六张数卡。而每张数卡的第一个数（也就是最小的）则分别是二进制的 1，10，100，1000，10000，100000，即为 1，2，4，8，16，32. 如果某个数在二进制表达下某些位数是 1，那么它就会出现在对应的数卡上。此时这些数卡上的首位数字相加则恰好就是这个数。例如 1011 会出现在以 1，10，1000 为首的数卡上，则 1011 也可写成 1+10+1000，因此满足了 $\forall x \in \mathbb{Z}, 1 \le x \le n$，所有包含 x 的数卡上的首位数（最小数）之和恰好为 t .

3.1.2 证明：当数卡张数为 k 时，识别量 n 的最大值是 $2^k - 1$.

根据二进制编制方法，显然可以用 k 张卡片表示出 $1,10,11,100,\cdots\cdots,\underbrace{1111\cdots111}_{k个1}$ 这 $2^k - 1$ 个数（以及什么卡片都不取的 0）

其次，每种卡片有取和不取两种取法，那么 k 张卡片就有 2^k 个取法，除去什么都没取表示 0 以外，还有 $2^k - 1$ 种取法，因此至多表示 $2^k - 1$ 个数字，因此 $n \le 2^k - 1$.

综上所述，n 的最大值是 $2^k - 1$

3.1.3 基于二进制原理的单面数卡模型的检验（以题目中 $n = 60$ 为例）

以 31 为例验证：

31 可以用二进制表示为 11111，应该出现在前五张卡片上。

同样，$31 = 1 + 2 + 4 + 8 + 16$，满足条件。具体原题中的数卡及相应二进制数卡在附录中体现。

3.2 对于问题 2 的解答

3.2.1 说明对于识别量为 80 的数卡，有四张

与上文同理，若有 k 张数卡，则 $n \le 3^k - 1$. 因此，至少需要四张。而四张是可行的。我们这样考虑：

把 1，2，3，…，80 根据三进制表示成 1，2，10，11，12，20，…，1111. 末尾为 1 的写在第一张卡片的正面，末尾为 2 的写在第一张卡片的反面，末尾为 0 的就不写。用同样的方法根据倒数第二位是 1 还是 2 还是 0 确定是否写在第二张卡片以及写在第二张卡片的正面还是反面。

这样一来这四张卡片的开头（即最小数）便是 1 和 2，3 和 6，9 和 18，54 和 27. 并且每个数字会根据三进制拆分为这 8 个数字中的某些数字的和，从而出现在对应的卡片的正面或反面上。这样的做法是可取的。

3.2.2 证明不计顺序的识别量为 80 的双面数卡的唯一性

证明：我们用归纳法来证明这一事实。

考虑用m张卡片来识别$1\sim 3^m-1$这些数。下证卡片上数字的安排方法不计卡片顺序和正反面互换是唯一的。

当$m=1$时，显然只有正反面分别写 1 和 2 这一种方法。

若$m=k-1$时成立，下证$m=k$时也成立。

首先我们知道，这k张纸片一共有3^k种取法。我们把什么都不取这一种取法记作识别数字 0，此时一共有3^k种选取方法。而需要表示的数也是3^k个（包含 0）。而一个选法只对一个数字，因此选法和数字之间的映射是一个双射。每一种选法都和一个数字一一对应。

考虑 1 所在的卡片，我们把它记作卡片 A，假设 A 没有 1 的另一面的最小数是$t\left(t\geq 2\right)$。下证$t=2$.

不取卡片 A 的取法一共有3^{k-1}种，这对应了3^{k-1}个数字我们把它们记为$y_1,y_2,\cdots,y_{3^{k-1}}(y_1<y_2<y_3<\cdots<y_{3^{k-1}})$. 在这些取法的基础上，再多取一个卡片 A 有 1 或 t 的那一面，那么就对应了另外$2\times 3^{k-1}$个数字分别是$y_1+1,y_2+1,\cdots,y_{3^{k-1}}+1$和$y_1+t,y_2+t,\cdots,y_{3^{k-1}}+t$. 由于映射是双射，因此这些数都是不同的。

那么我们有$\{y_1,y_1+1,y_1+t,y_2,y_2+1,y_2+t,\cdots,y_{3^{k-1}},y_{3^{k-1}}+1,y_{3^{k-1}}+t\}=\{0,1,2,\cdots,3^k-1\}$，由$y_1<y_2<y_3<\cdots<y_{3^{k-1}}$及$0<1<t$知左边集合中第二大的数一定是$y_{3^{k-1}}+1$或者$y_{3^{k-1}-1}+t$. 而显然第二大的数是$y_{3^{k-1}}+t-1$. 然而$y_{3^{k-1}-1}<y_{3^{k-1}}$，因此$y_{3^{k-1}-1}+1\leq y_{3^{k-1}}$，又由于这些数两两不同。于是$y_{3^{k-1}-1}+1<y_{3^{k-1}}$，有$y_{3^{k-1}-1}+t<y_{3^{k-1}}+t-1$，于是$y_{3^{k-1}}+1=y_{3^{k-1}}+t-1$，所以$t=2$.

于是$\{y_1,y_1+1,y_1+2,y_2,y_2+1,y_2+2,\cdots,y_{3^{k-1}},y_{3^{k-1}}+1,y_{3^{k-1}}+2\}=\{0,1,2,\cdots,3^k-1\}$，可得$y_i=3i-3(i=1,2,\cdots,3^{k-1})$.

此时考虑剩下的$k-1$张卡片，对它们的选取需要得到$0,3,6,\cdots,3^k-3$. 而根据归纳，$k-1$张卡片识别$0,1,2,\cdots,3^{k-1}-1$这3^{k-1}个数的方法是唯一的。又由于映射$f:3y\to y(y=0,1,2,\cdots 3^{k-1}-1)$在加法上是同构的。因此识别$0,3,6,\cdots,3^k-3$的方法也是唯一的。

因此若$m=k-1$时成立，则$m=k$时也成立。

综上所述，用k张卡片来识别$1\sim 3^k-1$这些数，卡片上数字的安排方法不计

卡片顺序和正反面互换是唯一的。

3.3　对于问题 3 的解答及模型的建立

由于$3^4-1<100<160<3^5-1$，因此对于两种不用的n都需要 5 张卡片。

3.3.1　证明$n=100$时$r\geq 25$，$n=160$时$r\geq 46$.

证明：$n=100$时，计$r_1,r_2,\cdots r_{10}$为卡片 10 面的数字个数，而$x_1,x_2,\cdots x_{10}$是 10 面的首位数（即最小数，其中$x_{2i-1}\neq x_{2i}$，它们是第i张卡片的正反面）。则$r_i\leq r(k=1,2,3,\cdots,10)$，于是有$\sum_{k=1}^{10}r_i\leq 10r$，当且仅当$r_i(i=1,2,\cdots,10)$都相等时取等号。然而把这 100 个不同的数字都用$x_1,x_2,\cdots x_{10}$表达出来，用一个数的方法至多$2C_5^1=10$个，两个数的方法至多$2^2C_5^2=40$个，三个数的方法至多有$2^3C_5^3=80$个，因此至少需要用$10+40\times 2+50\times 3=240$个数。因此$240\leq\sum_{k=1}^{10}r_i\leq 10r$，于是$24\leq r$，而如果取等号，则$5050=\sum_{i=1}^{100}i=\sum_{i=1}^{10}x_ir_i=r\sum_{i=1}^{10}x_i=24\sum_{i=1}^{10}x_i$，而 5050 不是 24 的整数倍，矛盾。故$r\geq 25$.

$n=160$时，另由于四个数的方法至多有$2^4C_5^4=80$个，因此至少需要$10+40\times 2+80\times 3+30\times 4=450$个数，因此$450\leq\sum_{i=1}^{10}r_i\leq 10r$，于是$45\leq r$，同理取等号时有$5050=\sum_{i=1}^{100}i=\sum_{i=1}^{10}x_ir_i=r\sum_{i=1}^{10}x_i=45\sum_{i=1}^{10}x_i$，而 5050 不是 45 的整数倍，矛盾。故$r\geq 46$.

3.3.2　寻找具体的数卡的解

以$n=100$为例，我们寻找数卡的过程分成两件事情来做。

一、找到比较好的数组$(x_1,x_2,\cdots x_{10})$的分布范围

我们在问题二的背景下，我们发现用三进制是一个很好的做法。然而，如果把类似的做法搬过来，确实可以表示出这些数，但由于 1-100 在三进制下各个位数上的数字 1，2，0 的分布是不均匀的，因此这样得到的$r_i\left(i=1,2,\cdots,10\right)$并不均匀。所以由于 81，162 这样的数字太大，我们把其中比较大的x_k变小。而小的

x_k保留以确保数字加法之间的灵活性。然而根据上面的估计我们有，

$5050=\sum_{k=1}^{100}k=\sum_{k=1}^{10}x_k r_k \le r\sum_{k=1}^{10}x_k$，因此我们有$\frac{5050}{r}\le\sum_{k=1}^{10}x_k$，想要让$r$尽量小，就需要让$\sum_{i=1}^{10}x_i$尽量大，因此我们不能把大的数字放的太小，同时我们尽量不让某个

由于考虑到下一步的复杂度，我们在这一步尽量把范围压缩到比较小。在保证两两之和以及三个数的和尽量不相通的原则下，适当计算之后给出以下范围：

$x_1=1,x_2=2,x_3=3,x_4=6,x_5=9,16\le x_6\le 17,22\le x_7\le 25,32\le x_8\le 37$

$44\le x_9\le 51,56\le x_{10}\le 64$.

二、使用程序通过算法，对固定的某个数组$(x_1,x_2,\cdots x_{10})$计算得到尽量优化的数卡的解，得到r能达到的最小值。

先来考察优化出的解需要满足的条件。为了让$\frac{5050}{r}\le\sum_{i=1}^{10}x_i$这个不等式尽量接近等号，我们就需要让$r_i\left(i=1,2,\cdots,10\right)$尽量接近。而另一方面，由于$r=\max\{r_1,r_2,\cdots,r_{10}\}$，我们希望每个$r_i\left(i=1,2,\cdots,10\right)$都尽量小。

现在我们来讲述得到数卡的解并且优化到r更小的情况的算法。

Step1 根据给定的十个数来计算它们作一次或多次加法可以得到的所有和（包括本身）。把这些加法方式和得到的结果（也就是和）按计算顺序记下来（自动剔除和超过 100 的）。

Step2 把这些式子和对应的结果从前往后进行第一次读取。每次读取进行如下操作：

1：如果是第一次读取到这个大小的结果，那么把这样的加法方式记录在虚拟的 10 张数卡（存储空间）上，例如对于 6+9=15，就把 15 这个数记录在虚拟数卡上以 6 和 9 开头的数卡上。

2：如果在这次之前。已经读取到了一样大小的结果。那么先计算虚拟数卡上即时的$r_k\left(k=1,2,\cdots,10\right)$也就是每一面数卡上的数字个数以及$r$，然后比较这两种加法，如果新的一组更好，就把原来的那组替换掉，否则就什么都不做。

关于“更好”的判定方法如下：

①计算如果换成新的哪一种加法方式之后，把对应的原来的加法方式的进行加法运算的数字在数卡上删去，把换上去的方式的进行加法的数字替换上去。再去计算即时的$r_k\left(k=1,2,\cdots,10\right)$，得到即时的新的$r$，与原来的$r$比较，如果变小了，则视为更好。

②如果两种方法的r相同，则把$\sum^{10} r_i$更小的那一组视为更好。

③如果两种方法的$\sum_{i=1}^{10} r_i$仍然相同，则计算两种方法的$r_i\left(i=1,2,\cdots,10\right)$这十个数的方差，方差更小的视为更好。

④如果以上三次比较都相同，则任取一种。

Step3　进行完前两个步骤后，已经得到一种数卡。为了让数卡的r更小，重复 Step2 若干次。

（这里具体的重复次数可以设定。初始化为 9 次。即一共进行十次 step2）

在重复次数足够大之后，得到的答案逐渐趋于稳态，不会再改变。

现在我们把上面两个步骤结合起来。对给定范围的一些$(x_1,x_2,\cdots x_{10})$的值遍历计算之后优化得到数卡的解，并最终给出我们能给出的最小值。

在这个限制下。我们得到了 25 种不同$(x_1,x_2,\cdots x_{10})$构成的不同数卡，得到的r最小值为 25.而上文已经证明 25 是理论下界。因此这个最小值已经是极限。

在此给出两种。其余见附录。

数卡集 A

卡 1 正面　数字个数：25

1 4 7 10 13 17 20 23 26 29 33 36 39 42 45 49 51 57 60 62 68 71 78 94 100

卡 1 反面　数字个数：25

2 5 8 11 14 18 21 24 27 30 34 37 40 43 46 52 55 58 63 74 79 84 85 90 95

卡 2 正面　数字个数：25

3 4 5 12 13 14 19 20 21 25 26 27 35 36 37 45 53 55 64 69 73 75 80 86 96

卡 2 反面　数字个数：25

6 7 8 15 24 28 29 30 38 40 44 46 47 54 56 57 58 65 67 68 76 85 87 89 97

卡 3 正面　数字个数：25
9 10 11 12 13 14 15 31 41 42 43 45 47 59 60 65 70 71 73 76 81 87 91 92 97

卡 3 反面　数字个数：25
16 17 18 19 20 21 24 39 44 46 48 49 54 66 69 77 78 79 80 85 88 90 98 99 100

卡 4 正面　数字个数：25
22 23 25 26 27 28 29 30 31 39 44 46 72 74 75 81 83 86 87 88 89 90 92 99 100

卡 4 反面　数字个数：25
32 33 34 35 36 37 38 40 41 42 43 45 47 48 49 54 82 84 91 93 94 95 96 97 98

卡 5 正面　数字个数：25
50 51 52 53 55 56 57 58 59 60 65 66 69 72 74 75 81 82 84 87 88 90 91 97 98

卡 5 反面　数字个数：25
61 62 63 64 67 68 70 71 73 76 77 78 79 80 83 85 86 89 92 93 94 95 96 99 100

数卡集 B

卡 1 正面　数字个数：25
1 4 7 10 13 17 20 23 25 28 31 36 45 49 52 55 58 61 64 73 76 80 86 88 100

卡 1 反面　数字个数：25
2 5 8 11 14 18 21 26 29 34 37 42 43 50 53 56 59 62 65 68 71 74 77 94 97

卡 2 正面　数字个数：25
3 4 5 12 13 14 19 20 21 27 28 29 35 36 37 44 45 55 56 66 68 70 82 90 98

卡 2 反面　数字个数：24
6 7 8 15 22 23 30 31 38 39 46 47 54 57 58 59 69 71 78 81 85 86 89 93

卡 3 正面　数字个数：25
9 10 11 12 13 14 15 33 39 41 43 44 45 47 60 61 62 72 73 74 78 84 92 94 96

卡 3 反面　数字个数：25
16 17 18 19 20 21 22 23 40 42 46 48 49 50 54 67 70 79 80 82 85 86 91 99 100

卡 4 正面　数字个数：24
24 25 26 27 28 29 30 31 33 39 40 42 46 75 76 77 81 84 87 88 90 91 93 96

卡 4 反面　数字个数：24
32 34 35 36 37 38 41 43 44 45 47 48 49 50 54 83 89 92 94 95 97 98 99 100

卡 5 正面　数字个数：25
51 52 53 55 56 57 58 59 60 61 62 67 70 75 76 77 81 83 84 89 91 92 94 99 100

卡 5 反面　数字个数：24
63 64 65 66 68 69 71 72 73 74 78 79 80 82 85 86 87 88 90 93 95 96 97 98

而我们证明了，因此我们已经取到了最小值。

对于 $n=160$，使用同样的方法。我们给出以下的数组 $(x_1,x_2,\cdots x_{10})$ 的范围：

$x_1=1,x_2=2,x_3=3,x_4=6,x_5=9,x_6=18,x_7=27$，

$38\le x_8\le 54,60\le x_9\le 80,90\le x_{10}\le 109$

最后我们得到了 48 组解，得到的 r 的最小值为 48，而理论下界为 46，可是我们并不能得到 r 为 46 或 47 的数卡的解。

在此给出两组，其余见附录。

数卡集 C

卡 1 正面　数字个数：47
1 4 7 10 13 16 19 22 25 28 31 34 37 39 46 48 54 57 60 63 66 69 72 75 78 81 84 87 90 93 96 99 102 104 109 115 122 124 130 133 136 139 142 144 147 151 159

卡 1 反面　数字个数：46
2 5 8 11 14 17 20 23 26 29 32 35 40 43 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91 94 97 100 107 116 119 125 128 131 134 137 140 145 148 154 157 160

卡 2 正面　数字个数：48
3 4 5 12 13 14 21 22 23 30 31 32 41 43 50 52 59 60 61 68 69 70 77 78 79 86 87 88 95 96 97 106 108 109 113 117 119 126 128 135 136 137 146 147 148 153 155 157

卡 2 反面　数字个数：48
6 7 8 15 16 17 24 25 26 33 34 35 42 44 51 53 54 55 62 63 64 71 72 73 80 81 82 89 90 91 98 99 100 111 118 120 127 129 130 131 138 139 140 149 156 158 159 160

卡 3 正面　数字个数：48
9 10 11 12 13 14 15 16 17 36 37 42 47 48 49 50 52 53 54 55 74 75 76 77 78 79 80 81 82 101 102 112 114 115 116 117 118 119 120 141 142 152 154 155 157 158 159 160

卡 3 反面　数字个数：47
18 19 20 21 22 23 24 25 26 45 46 51 56 57 58 59 60 61 62 63 64 83 84 85 86 87 88 89 90 91 110 113 121 122 123 124 125 126 127 128 129 130 131 150 151 153 156

卡 4 正面　数字个数：43
27 28 29 30 31 32 33 34 35 36 37 42 45 46 51 92 93 94 95 96 97 98 99 100 101 102 110 113 132 133 134 135 136 137 138 139 140 141 142 150 151 153 156

卡 4 反面　数字个数：45
38 39 40 41 43 44 47 48 49 50 52 53 54 55 56 57 58 59 60 61 62 63 64 103 104 106 112 118 121 122 127 143 144 145 146 147 148 149 152 154 155 157 158 159 160

卡 5 正面　数字个数：48
65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 106 110 112 113 118 121 122 127

卡 5 反面　数字个数：48
105 107 108 109 111 114 115 116 117 119 120 123 124 125 126 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160

数卡集 D

卡 1 正面　数字个数：48
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 54 57 60 63 66 69 72 75 78 81 84 87 90 94 97 100 103 116 119 127 131 134 137 139 142 144 147 150 153 156 159

卡 1 反面　数字个数：48
2 5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 52 55 58 61 64 67 70 73 76 79 82 85 88 91 95 101 104 106 109 113 114 122 125 128 132 140 145 148 151 154 157 160

卡 2 正面　数字个数：47
3 4 5 12 13 14 21 22 23 30 31 32 39 40 41 48 49 53 54 55 65 66 67 76 83 84 85 92 96 97 105 110 115 116 123 124 125 133 134 141 142 146 147 148 155 156 157

卡 2 反面　数字个数：48
6 7 8 15 16 17 24 25 26 33 34 35 42 43 44 51 56 57 58 68 74 75 77 78 79 86 87 88 99 100 101 106 108 117 118 119 126 127 128 135 136 137 149 150 151 158 159 160

卡 3 正面　数字个数：48
9 10 11 12 13 14 15 16 17 36 37 38 39 40 41 42 43 44 59 60 61 71 72 73 76 77 78 79 98 102 103 104 105 106 108 121 124 129 135 152 153 154 155 156 157 158 159 160

卡 3 反面　数字个数：46
18 19 20 21 22 23 24 25 26 45 46 47 48 49 51 69 70 74 75 80 81 82 83 84 85 86 87 88 107 109 110 111 113 117 130 131 132 133 134 136 137 138 139 140 141 142

卡 4 正面　数字个数：47
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 51 89 90 91 92 98 106 107 109 110 120 122 123 125 126 127 128 129 135 138 139 140 141 142

卡 4 反面　数字个数：48
50 52 53 54 55 56 57 58 59 60 61 69 70 74 75 112 114 115 116 118 119 121 124 130 131 132 133 134 136 137 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160

卡 5 正面　数字个数：47
62 63 64 65 66 67 68 71 72 73 76 77 78 79 80 81 82 83 84 85 86 87 88 89

90 91 92 98 106 107 109 110 112 114 115 116 118 119 121 124 130 131 132 133 134 136 137

卡 5 反面　数字个数: 48
93 94 95 96 97 99 100 101 102 103 104 105 108 111 113 117 120 122 123 125 126 127 128 129 135 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160

.4. 数卡模型的实际应用问题

4.1 基于二进制原理的单面数卡模型在二维码中的应用

4.1.1 背景介绍

目前，随着智能手机的普及，信息网络的发展，二维码逐渐走入人们的视野。无论是用二维码表示名片、地图等实现信息获取，还是网站跳转，广告推送，以及各种各样的优惠促销手机支付等，二维码都以其灵活便捷，成本低廉，且能够存储汉字、数字和图片等信息的多种优点影响着我们的生活。

4.1.2 二维码的原理及数卡模型在二维码中的应用

与本题中将正整数转换为二进制处理类似，二维码其实就是由很多 0、1 组成的数字矩阵，用某种特定的几何图形按一定规律在平面（二维方向上）分布的黑白相间的图形记录数据符号信息的。使用若干个与二进制相对应的几何形体来表示文字数值信息，每种码制有其特定的字符集；二维条码/二维码能够在横向和纵向两个方位同时表达信息，因此能在很小的面积内表达大量的信息。

而本题中的单面数卡也恰恰暗含着这样的思想，即可以用很少的卡片来表示大量的数据，表示大量的信息。

4.2 数卡模型解决密码学的问题

4.2.1 背景解释

信息安全如今的地位越来越重要，无论是日常的用 WiFi 上网，或是在网上进行交易，都很容易出现隐私安全泄露等问题。而密码学在解决信息安全的问题上更起举足轻重的作用。

4.2.2 尝试用数卡模型解决密码学的问题

针对双面数卡模型的第三、第五问，我们可以得到这样的启发，即当 $3^{k-1}-1<n\le 3^{k}$，且 n 处于中间位置时，即使给出 k 张数卡正反面的最小值，我们仍会得到相当多种数卡的表示形式。这意味着，如果把这 k 张数卡当做“公钥”，

对任意给定的一个数字，我们仍很难找到对应的正确数字，这样，我们可以认为用数卡模型解决密码学的问题是可行的，也是有前景的。

5. 关于 n 的延拓的思考（附加题）

5.1 对任意正整数n计算需要的卡片数和理论下界的值

由于在问题三中使用的下界计算方法算法在n比较小的时候具有普适性，因此我们可以

1、把对n计算卡片的个数k和理论下界r的方法延拓到整个正整数上。

2、把通过程序计算具体的数卡解答，并且得到目前优化方案下的最小值以及对应的数卡的方法延拓到$1 \sim 242$（也就是3^5-1）

下面介绍如何计算卡片的个数k和理论下界r。

1、首先我们知道k张卡片可以也至多识别3^k-1个数（问题 1 已证）因此如果用了k张卡片，必须也只需满足$n \le 3^k-1$，由题意方程$n \le 3^k-1$对于k的最小整数解就是卡片的个数。而$n \le 3^k-1 \Leftrightarrow \log_3(n+1) \le k$，因此卡片的个数也就是不小于$\log_3(n+1)$的最小整数。

2、其次我们给出计算理论下界r的方法，原理与问题三的做法一致。在此主要介绍算法。

在得到k的具体数值后，计算$\sum_{i=1}^{2k} r_i$的理论最小值。考虑使用t个数做加法有$2^t C_k^t$种方法。令$S_p = \sum_{i=1}^{p} 2^i C_k^i$，求出满足$S_p < n \le S_{p+1}$的$p$，现在我们再来计算$S = (t+1)(n-S_p) + \sum_{i=1}^{p} 2^i i C_k^i$，我们有$\sum_{i=1}^{2k} r_i \ge S$，于是有$2kr = \sum_{i=1}^{2k} r \ge \sum_{i=1}^{2k} r_i \ge S$，得到$r \ge \frac{S}{2k}$，再考虑能否取到等号。取到等号时必须满足$2k \mid S$且$\frac{S}{2k} \mid \frac{n(n+1)}{2}$，也就是$S \mid kn(n+1)$.

综上，若$2k \mid S$且$S \mid kn(n+1)$，则可以得到下界$\frac{S}{2k}$，否则便得到下界$[\frac{S}{2k}]+1$（这里的[]为取整函数）

5.2 对于比较小的n给出较优解

最后我们介绍对于比较小的n如何用程序得到目前优化方案下的最小值和数卡。当n处于 1 到 242 时，我们只需要用 5 张以内的卡片即可。

$1 \le n \le 8$时可以直接得到答案：

n=1	卡片 A 正面 1 反面 无	
n=2	卡片 A 正面 1 反面 2	
n=3	卡片 A 正面 1 反面 无	卡片 B 正面 3 反面 无
n=4	卡片 A 正面 1 反面 2	卡片 B 正面 3 反面 4
n=5	卡片 A 正面 1 反面 2,5	卡片 B 正面 3,5 反面 4
n=6	卡片 A 正面 1,4 反面 2,5	卡片 B 正面 3,4,5 反面 6
n=7	卡片 A 正面 1,4,7 反面 2,5	卡片 B 正面 3,4,5 反面 6,7
n=8	卡片 A 正面 1,4,7 反面 2,5,8	卡片 B 正面 3,4,5 反面 6,7,8

$9 \le n \le 26$时需要用 3 张卡片；$27 \le n \le 80$时需要用 4 张卡片；$81 \le n \le 242$时需要用 5 张卡片。我们用和问题三中相同的算法对给定的数组$(x_1, x_2, \cdots x_{2k})$计算数卡的解和最小值。

我们对不同的n使用一个比单独的n更大的范围来确保在得到的数卡的解是当前优化方案能达到的最小值的同时这一范围对多个具有普适性。在对几个离散的n进行单独计算后，通过计算确定最后所用的范围。（计算结果见附录）

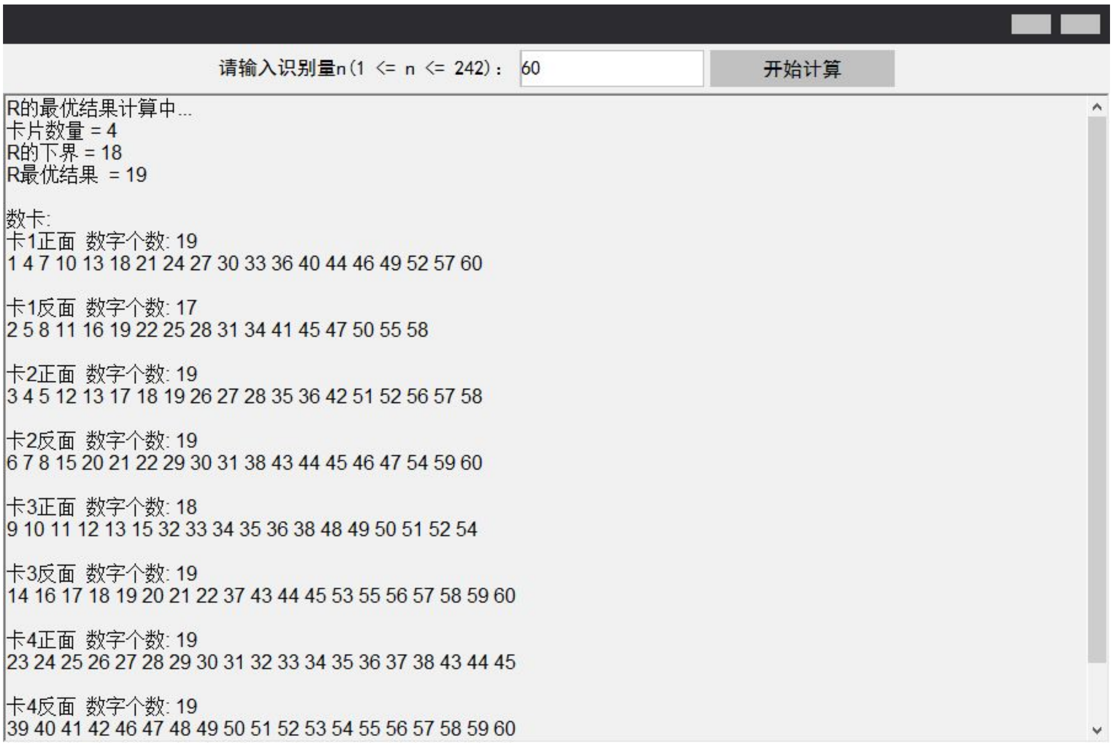
以下给出对不同的n给使用的$(x_1, x_2, \cdots x_{2k})$的范围。

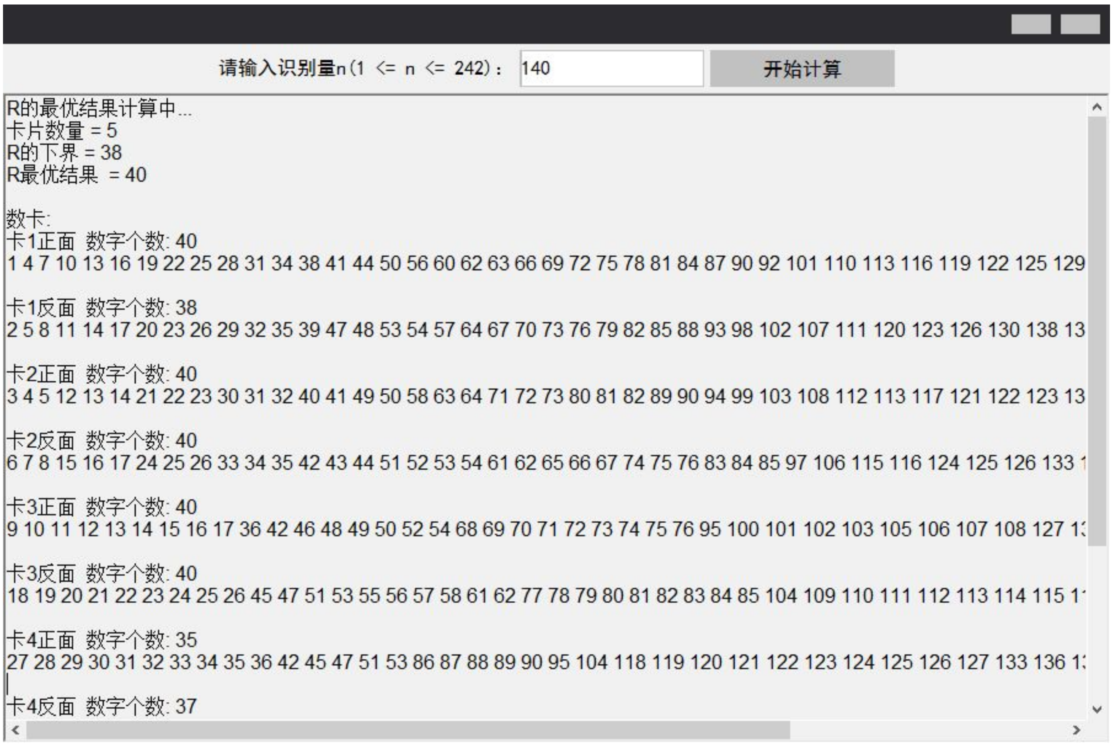
n的范围	9-26	27-80	81-120	121-161	162-242
x_1的范围	1	1	1	1	1
x_2的范围	2	2	2	2	2
x_3的范围	3	3	3	3	3
x_4的范围	4-6	4-6	6	6	6
x_5的范围	6-9	6-9	9	9	9
x_6的范围	10-18	10-18	15-17	18	18
x_7的范围	*	14-27	22-26	27	27
x_8的范围	*	28-54	28-42	36-54	54
x_9的范围	*	*	36-62	55-81	70-81
x_{10}的范围	*	*	46-75	67-110	96-162

（若出现$x_i \ge x_{i+1}$则直接跳过这个数组）

具体的程序见附录。

以下是程序运行截图：





6. 模型的评价

这个模型主要是在对数组$(x_1, x_2, \cdots x_{10})$进行一定限制的遍历之后进行逐步调整。

6.1 对第一步的优缺点评价：

就第一步的做法而言，有一定的缺点，就是在n比较大的时候，无法得到具体的限制，使得程序无法自动进行；并且在n增大时，遍历的总次数会增大的更快。（例如变成n原来的 3 倍时，会使得总次数变成原来的数十倍）但是在n比较小的时候，确实可以从完全遍历把范围压缩很多。

6.2 对第二步的优缺点评价：

就第二步而言。优点在于把 2 的数十次以上方幂，或者 3 的数十次以上方幂次大运算（如果遍历的话）转化为只要数百次大运算（把幂运算改成了加法运算），使得计算速度加快，保证了我们对任意的比较小的n，都可以在数分钟内得到类似 25，48 这样的已经比较好的最小值。然而缺点在于最终也得到的是接近最优解的解，我们也不能确定我们得到的解是最优的。

参考文献：

[1]密码学原理与实践（第三版）Stinson，D.R 著 冯登国　等译

[2]二维码的概念与分类 中国物联网[引用日期 2015-01-12]