３分シャフリングの研究

三浦 拓馬
学習院大学 理学部 数学科

平成 24 年 2 月 16 日

目 次

1　目的

N 枚のカードを 3 分割し, シャッフルした時の結果を調べ, その周期等の性質を考察する.

1.1　「シャフリング」について

そもそも,「シャフリング」とは何なのかを簡単に説明したい.

一番身近な例としてトランプがある.
（なので, この論文の説明も「〜〜枚のシャフリング」という言い方を使っていく）

トランプの混ぜ方にも幾つかあるが, ここで引き合いに出すのは,
２つに分けたトランプを交互に入れ込んでいく混ぜ方である.

例で示す.
下の $\boxed{}$ の列を, トランプの山とする.（番号は上から何番目のカードかを示す.）
$\boxed{1}, \boxed{2}, \boxed{3}, \cdots, \boxed{52}$

これを２つに分割する.
$\boxed{1}, \boxed{2}, \boxed{3} \cdots \boxed{51}, \boxed{26}$
$\boxed{27}, \boxed{28}, \cdots, \boxed{52}$

パラパラと, 交互に入れ込んでいく.
$\boxed{1}, \boxed{27}, \boxed{2}, \boxed{28}, \cdots, \boxed{51}, \boxed{26}, \boxed{52}$
この一連の操作を「シャフリング」と呼ぶ.
特に上の例は, 山を２つに分割した上でのシャフリングなので「２分シャフリング」という.

これから論じるのは, 山を３つに分割したシャフリングである.

2　方法

2.1　3 分シャフリングの方法

ここでは, 全体の枚数 N は 3 の倍数であるとする

$L_0 = [1, 2, 3, \cdots, n-1, n, n+1, \cdots, 2n-1, 2n, 2n+1, \cdots 3n-2, 3n-1, 3n]$
この山を 3 等分にし, それぞれの山に A,B,C と名付ける.
$L_0 = [[1, 2, 3, \cdots, n](=A), [n+1, n+2, \cdots, 2n](=B), [2n+1, 2n+2, \cdots, 3n](=C)]$
C,B,A の順に上から並べる.

$$C = \begin{bmatrix} 2n+1 & ,2n+2 & ,2n+3 & ,\ldots & ,3n \end{bmatrix}$$
$$\mathrm{B} = \begin{bmatrix} n+1 & ,n+2 & ,n+3 & ,\ldots & ,2n \end{bmatrix}$$
$$\mathrm{A} = \begin{bmatrix} 1 & ,2 & ,3 & ,\ldots & ,n \end{bmatrix}$$

そして,C の先頭,B の先頭,A の先頭,C の 2 番目,B の 2 番目 $\cdots$ という風に並べる.
$L_1 = [2n+1, n+1, 1, 2n+2, n+2, 2, 2n+3, \cdots, 3n, 2n, n]$

この一連の操作を「3 分シャフリング」という.
この操作を, 元の配列 L_0 に戻るまで繰り返していく.

2.2　補充を含む 3 分シャフリングの方法

ここからは, 必ずしも N は 3 の倍数であることを想定しない.

N を 3 で割る.

- $N = 3k(k \in \mathbb{N})$ の時

  上の方法を使えばいい.

- $N = 3k+2$ または $N = 3k+1(k \in \mathbb{N})$ の時

  それぞれ 1 枚,2 枚を補充して,3 の倍数に揃えて上の方法を使って混ぜる.
  その後, 補充したカードを除いて間を詰めた配列を, シャッフル後の配列とする.

この一連の操作を「補充付き 3 分シャフリング」という.

・$N=3k+1(k\in\mathbb{N})$ の時

$L_0=[1,2,3,\cdots,k,\cdots 2k,\cdots,3k,3k+1]$

上の設定から, 仮のカードとして$\boxed{\mathrm{A}}$,$\boxed{\mathrm{B}}$というカードを補充する.

$L_0'=[1,2,3,\cdots,k,\cdots 2k,\cdots,3k,3k+1,\boxed{\mathrm{A}},\boxed{\mathrm{B}}]$

これにより,L_0' の枚数は 3(k+1) 枚となるため 3 分シャフリングを実行できる.

$L_1'=[2k+3,k+2,1,2k+4,k+3,2,\cdots,3k+1,2k,k-1,\boxed{\mathrm{A}},2k+1,k,\boxed{\mathrm{B}},2k+2,k+1]$

仮に補充した$\boxed{\mathrm{A}}$,$\boxed{\mathrm{B}}$を取り除き, 間を詰める.

$L_1=[2k+3,k+2,1,2k+4,k+3,2,2k+5,k+4,3,\cdots,3k+1,2k,k-1,2k+1,k,2k+2,k+1]$

・$N=3k+2(k\in\mathbb{N})$ の時

A の部分に数が入っているので, 同様の操作を行った後に B のみを取り除けばいい.

この操作を, 元の配列 L_0 に戻るまで繰り返す.

(例)7 枚の場合

$L_0=[1,2,3,4,5,6,7]$

$7=3\times 2+1$ と表せるので, 仮のカードとして$\boxed{\mathrm{A}}$,$\boxed{\mathrm{B}}$というカードを補充する.

$L_0'=[1,2,3,4,5,6,7,\boxed{\mathrm{A}},\boxed{\mathrm{B}}]$

シャッフルする

$L_1'=[7,4,1,\boxed{\mathrm{A}},5,2,\boxed{\mathrm{B}},6,3]$

補充したカード$\boxed{\mathrm{A}}$,$\boxed{\mathrm{B}}$を取り除き, 間を詰める

$L_1=[7,4,1,5,2,6,3]$

これをシャッフル後の配列とする.

この操作を, 元の配列 L_0 に戻るまで繰り返していく.

しかし, 定義通りに事を進めると,

カードを追加するプログラムと, 排除するプログラムを別個に組む事になり, 手間がかかる.

〜・〜

そこで, シャッフルの作業についてもう少し煮詰め, 作業の簡略化を図る.

1 回シャッフルする作業について.

$L_0=[1,2,3,\cdots,n-1,n,n+1,\cdots,2n-1,2n,2n+1,\cdots 3n-2,3n-1,3n]$

L_0 を 3 等分にして, 左のパートをA, 中央をB, 右をCとする.

さらに, それぞれのパートの要素に,1,2,3 ‥ と番号付けを行う.

すると, 以下のように表せる.

$\tilde{L}_0=[a_1,a_2,a_3,\cdots,a_n,b_1,b_2,\cdots,b_n,c_1,c_2\cdots,c_n]$

これをシャッフルすると,

$\tilde{L}_1=[c_1,b_1,a_1,c_2,b_2,a_2,\cdots,c_{n-1},b_{n-1},a_{n-1},c_n,b_n,a_n]$

となる.

前から 3 つずつリストに区切ってみると,

$\tilde{L}_1=[[c_1,b_1,a_1],[c_2,b_2,a_2],\cdots,[c_{n-1},b_{n-1},a_{n-1}],[c_n,b_n,a_n]]$

という風になる.

この様にみると,3 分シャフリングは,
元の配列を 3 等分に分けて, それぞれのリストの先頭を右から取っていって並べる作業を繰り返せばいいとわかる.
また, 補充カードとなる$\boxed{\mathrm{A}}$,$\boxed{\mathrm{B}}$は, それぞれ c_{n-1},c_n に入ってくる.

そこで, 補充をした際は, 此処に空の要素が入るという事を予め想定し, 対処する様なプログラムを組み, シャフリングのプログラムに付加させれば, 簡略を図ることができる.

また $\tilde{L}_1$ から,3 分シャッフルは全単射の写像である.

2.3 プログラム

最後に,3 分シャフリング, 及び補充付き 3 分シャフリングを実現するためのプログラムを提示する.

〜・〜

```
append0(Z=[]+Z).
append0([A|Z]=[A|X]+Y):-append0(Z=X+Y).

left(A=[]+A,0):-!.
left(A=B+C,N):-N>0,N1 is N-1,!,
              left(A=B1+[X|C],N1),append0(B=B1+[X]),!.

power_aux(N,W,K=X^M mod P):-N<M,N1 is N+1,
                          W1 is (X*W) mod P,power_aux(N1,W1,K=X^M mod P).
power(K=X mod P):-power_aux(0,1,K=X^1 mod P).

unit(1,[1]).
unit(N,L):-N>1,N1 is N-1,
    unit(N1,L1),append0(L=L1+[N]).

r_unit(1,[1]).
r_unit(N,L):-N>1,N1 is N-1,
    r_unit(N1,L1),append0(L=[N]+L1).

left31(L,A,B,C):-length(L,N),power(K=N mod 3),
                (K=:=1 ->(N1 is (N+2)/3,left(L=A+D,N1),left(D=B+C,N1));
                 K=:=2 ->(N1 is (N+1)/3,left(L=A+D,N1),left(D=B+C,N1));
                 N1 is N/3,left(L=A+D,N1),left(D=B+C,N1)),!.

shuf3([]+[]+[]=[]).
shuf3([X]+[]+[]=[X]).
```

```
shuf3([X|D]+[Y|C]+[]=[Y,X|M]):-shuf3(D+C+[]=M).
shuf3([X|D]+[Y|C]+[Z|A]=[Z,Y,X|M]):-shuf3(D+C+A=M).

sh3(L=M):-left31(L,A,B,C),shuf3(A+B+C=M).

shuffle3(M=N):-unit(N,L),sh3(L=M).

nshuf31(_,0,_):-!.
nshuf31(L,N,K):-length(L,S),unit(S,L1),sh3(L=M),
               N2 is N-1,K1 is K+1,
              (L1 \== M ->(write(K1),tab(1),write(M),nl,nshuf31(M,N2,K1));
               write(K1),tab(1),write(M),tab(1),write(l=K1)).

nshuf32(_,0,_):-!.
nshuf32(L,N,K):-length(L,S),unit(S,L1),r_unit(S,L2),sh3(L=M),
           N2 is N-1,K1 is K+1,
           (L2 == M ->(write(K1),tab(1),write(M),tab(1),write(reversed),nl,nshuf32(M,N2,K1));
            (L1 \== M ->(write(K1),tab(1),write(M),nl,nshuf32(M,N2,K1));
           write(K1),tab(1),write(M),tab(1),write(l=K1))).

nf3(N,M):-unit(N,L),tab(2),write(L),nl,nshuf32(L,M,0).

nshuf33(_,0,_):-!.
nshuf33(L,N,K):-length(L,S),unit(S,L1),r_unit(S,L2),sh3(L=M),
           N2 is N-1,K1 is K+1,
           (L2 == M ->(write(reversed=K1),nl,nshuf33(M,N2,K1));
            (L1 \== M ->(nshuf33(M,N2,K1));write(l=K1))).

nf31(N,M):-unit(N,L),nshuf33(L,M,0).

long_nf31(N,M):-write(N),tab(3),nf31(N,100000),nl,P is N+1,
        (P == M -> (write(P),tab(3),nf31(M,100000));long_nf31(P,M)).
```

3　結果

3.1　シャッフルの周期等について

まず, 枚数が 3 の倍数の時を考える.

一度だけシャッフルする

```
?-sh3([1,2,3,4,5,6,7,8,9]=M).
M = [7,4,1,8,5,2,9,6,3]
```

これを, 元の配列になるまで繰り返していく

シャッフルを連続して行うプログラムを用いる.

```
 ?- nshuf3([1,2,3,4,5,6,7,8,9],10,0).
1 [7, 4, 1, 8, 5, 2, 9, 6, 3]
2 [9, 8, 7, 6, 5, 4, 3, 2, 1]（「逆転」が生じている）
3 [3, 6, 9, 2, 5, 8, 1, 4, 7]
4 [1, 2, 3, 4, 5, 6, 7, 8, 9]（元の配列に戻った. 初めて戻った時の数を「周期」と呼ぶ）
5 [7, 4, 1, 8, 5, 2, 9, 6, 3]（以降は繰り返し）
6 [9, 8, 7, 6, 5, 4, 3, 2, 1]
7 [3, 6, 9, 2, 5, 8, 1, 4, 7]
8 [1, 2, 3, 4, 5, 6, 7, 8, 9]
9 [7, 4, 1, 8, 5, 2, 9, 6, 3]
10 [9, 8, 7, 6, 5, 4, 3, 2, 1]
true
```

シャッフルを行う事は PC 上で実現できた.

更に, 特徴を端的に捉える為に次のプログラムを考える.
・L_0 に戻った時ストップする.
・逆転が生じた時は明示する.
といった情報を加えたのが次のプログラムである

```
 ?- nshuf32([1,2,3,4,5,6,7,8,9],10,0).
1 [7, 4, 1, 8, 5, 2, 9, 6, 3]
2 [9, 8, 7, 6, 5, 4, 3, 2, 1] reversed
3 [3, 6, 9, 2, 5, 8, 1, 4, 7]
4 [1, 2, 3, 4, 5, 6, 7, 8, 9] l=4
true
```

逆転は頻繁には生じず,
多くの場合には逆転することなく元の配列に戻る.

```
 ?- nf3(15,10).
  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
1 [11, 6, 1, 12, 7, 2, 13, 8, 3, 14, 9, 4, 15, 10, 5]
2 [9, 2, 11, 4, 13, 6, 15, 8, 1, 10, 3, 12, 5, 14, 7]
3 [3, 6, 9, 12, 15, 2, 5, 8, 11, 14, 1, 4, 7, 10, 13]
4 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] l=4
true
```

枚数に因る分岐を含めたことで,3 の倍数でない場合も,
3 の倍数の時と同様に処理をすることができる.

```
 ?- nf3(10,15).
  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
1 [9, 5, 1, 10, 6, 2, 7, 3, 8, 4]
2 [8, 6, 9, 4, 2, 5, 7, 1, 3, 10]
3 [3, 2, 8, 10, 5, 6, 7, 9, 1, 4]
4 [1, 5, 3, 4, 6, 2, 7, 8, 9, 10]
5 [9, 6, 1, 10, 2, 5, 7, 3, 8, 4]
6 [8, 2, 9, 4, 5, 6, 7, 1, 3, 10]
7 [3, 5, 8, 10, 6, 2, 7, 9, 1, 4]
8 [1, 6, 3, 4, 2, 5, 7, 8, 9, 10]
9 [9, 2, 1, 10, 5, 6, 7, 3, 8, 4]
10 [8, 5, 9, 4, 6, 2, 7, 1, 3, 10]
11 [3, 6, 8, 10, 2, 5, 7, 9, 1, 4]
12 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] l=12
true
```

配列の表示を消去すれば, 単に逆転と周期を知りたい時に便利になる

```
 ?- long_nf31(3,10).
3   reversed=1   l=2
4   reversed=2   l=4
5   l=4
6   reversed=3   l=6
7   l=3
8   l=7
9   reversed=2   l=4
10   l=12
true
```

これを表に纏めたものが次のページの様になる.
(表中で, 逆転欄の◎は, 逆転に加え枚数と周期の一致が表れているものである.)
(周期に※印が付いているものについては, 考察のページにその理由が書かれている.)

枚数	周期	逆転
1	—	—
2	—	—
3	2	○
4	4	◎
5	4	○
6	6	◎
7	3	×
8	7	×
9	4	○
10	12	×
11	15	×
12	3	×
13	42	×
14	20	×
15	4	×
16	16	×
17	66	×
18	18	◎
19	65	×
20	45	×
21	5	×
22	48	×
23	68	×
24	20	○
25	170	×
26	66	×
27	6	○
28	28	×
29	138	×
30	30	◎
31	28	×
32	31	×
33	16	○
34	60	×
35	35	×
36	18	○
37	36	×
38	28	×
39	4	×
40	40	×

枚数	周期	逆転
41	390	×
42	42	◎
43	3740	×
44	231	×
45	11	×
46	528	×
47	1200	×
48	42	○
49	330	×
50	24	×
51	6	×
52	860	×
53	780	×
54	20	×
55	473	×
56	55	×
57	28	○
58	2610	×
59	190	×
60	10	○
61	1080	×
62	368	×
63	16	×
64	64	×
65	946	×
66	22	○
67	3828	×
68	276	×
69	12	×
70	※ 4620	×
71	276	×
72	12	○
73	468	×
74	630	×
75	18	○
76	76	×
77	222	×
78	78	◎
79	1288	×
80	120	×

枚数	周期	逆転
81	8	○
82	※ 11280	×
83	496	×
84	16	×
85	360	×
86	190	×
87	10	×
88	510	×
89	66	×
90	6	×
91	※ 19734	×
92	1035	×
93	23	×
94	445	×
95	1680	×
96	48	○
97	※ 16340	×
98	780	×
99	20	×
100	※ 12276	×
101	※ 2278	×
102	34	○
103	99	×
104	103	×
105	52	○
106	※ 2133	×
107	1431	×
108	27	×
109	※ 3036	×
110	204	×
111	12	×
112	※ 6732	×
113	※ 4884	×
114	44	×
115	※ 2465	×
116	※ 1653	×
117	29	×
118	140	×
119	45	×
120	5	×

3.2　巡回群への分解

３分シャフリングの枚数等の性質を更に究めるために,
ここで「サイクル分解」というものを定める.

> 元の配列を $L_0 = [1, 2, 3, \cdots, n]$,
> ３分シャフリングを１度行った配列を $L_1 = [\sigma(1), \sigma(2), \cdots, \sigma(n)]$ とする
>
> $L = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ \sigma(1) & \sigma(2) & \sigma(3) & \cdots & \sigma(n) \end{pmatrix}$ とし,
> 置換群 L を互いに素な巡回群の積の形へ分解する.
> こうして出来た巡回群の積
> $L = \tau_1\tau_2 \ldots \tau_t$
> を, 配列 L の「サイクル積」,τ_t をその「サイクル」と呼ぶ.
> また, この一連の操作を「サイクル分解」と呼ぶ.

また, これからのために１つ言葉を定める.

> $L = \begin{pmatrix} 1 & 2 & \ldots & t & \ldots & n \\ \sigma(1) & \sigma(2) & \ldots & \sigma(t) & \ldots & \sigma(n) \end{pmatrix}$
> として,$t = \sigma(t)$ だったとき,t はその置換群 L の「軸」と呼ぶ.

サイクル分解を行うプログラムを以下に示す.

〜・〜

```
delete1([A|L]-A=L).
delete1([B|L]-A=[B|M]):-delete1(L-A=M).
retract1(P):-retract(P),!.

collect1(c,L1):-collect(c,L),length(L,N),
      (N == 1 ->(L1=L);L1=L),!.
collect(c,[A|L]):-retract1(c(A)),collect(c,L),!.
collect(c,[]):-!.

cycle_g(F,A,[W,Z]):-delete1(F-(X:A)=L),
    assertz(c(A)),cycle_g(L,X,[W,Z]),!.
cycle_g(W,_,[W,D]):-collect1(c,D).

cycle_c([],C,C):-!.
cycle_c(W,[W|C],C):-!.
```

```
cycle0(F,F0,W):-member((_:A),F),cycle_g(F,A,[W,F0]),!.

cycle3([],[]).
cycle3(F,C1):-cycle0(F,W,F0),cycle3(F0,C0),cycle_c(W,C1,C0),!.


num_aux([],[],_).
num_aux([A|L],[(A:NI)|M],N):-length(L,I),NI is N-I,
                  num_aux(L,M,N).
num1(X,Y):-length(X,N),num_aux(X,Y,N),!.

cycleng([],[]).
cycleng([A|L],[S|M]):-length(A,S),cycleng(L,M),!.

gcd(_,0,_):-!.
gcd(A,B,C):-(A < B -> gcd(B,A,C));D is A mod B,
       (D == 0 ->(C=B);gcd(B,D,C)),!.
lcm(A,B,G):-gcd(A,B,C),G is (A*B)//C.
cyclcm(Nc,K,Q):-left(Nc=A+C,1),lcm(A,K,B),
         (C \== [] ->cyclcm(C,B,Q));lcm(Nc,K,Q).


cycle(N):-shuffle3(A=N),num1(A,R),cycle3(R,C),cycleng(C,Nc),
     cyclcm(Nc,1,Q),
     write(N),tab(1),write(C-Nc-Q),nl,!.
cycle(_).


long_cycle(N,M):-cycle(N),P is N+1,
        (P == M -> (cycle(M));long_cycle(P,M)).
```

特に使用するのは”cycle”という述語である.
これを用いれば,
・長さ N の L_0 をサイクル分解したもの
・サイクル τ_t のそれぞれの長さ
・τ_1 の長さから τ_t までの長さの最小公倍数
の 3 つを素早く表示できる.

(例)$N = 10$ の時

```
 ?- cycle(10).
10 [[1,9,8,3],[2,5,6],[4,10],[7]]-[4,3,2,1]-[12]
true.
```

これを表に纏めると次からのページの様になる.
ただ, 数が大きくなるとサイクル分解後の表示が長くなるので,$N = 30$ 以降は表示を省く.

また, 今回は軸を表示する形を採用した.
軸の表示を必要としない時は,
述語”collect1”中にある分岐後の括弧内の”L”の部分を”[]”に変えれば,
軸を除外した表示にできる.

```
 ?- cycle(8).
8 [[1,7,6,2,4,8,3]]-[7]-[7]
true.
```

表 1: 枚数とサイクル分解 その 1 (1〜30 枚)

枚数	サイクル分解	サイクルの型	LCM
2	[[1,2]]	[2]	[2]
3	[[1,3],[2]]	[2,1]	[2]
4	[[1,3,4,2]]	[4]	[4]
5	[[1,5,2,3],[4]]	[4,1]	[4]
6	[[1,5,4,6,2,3]]	[6]	[6]
7	[[1,7,3],[2,4,5],[6]]	[3,3,1]	[3]
8	[[1,7,6,2,4,8,3],[5]]	[7,1]	[7]
9	[[1,7,9,3],[2,4,8,6],[5]]	[4,4,1]	[4]
10	[[1,9,8,3],[2,5,6],[4,10],[7]]	[4,3,2,1]	[12]
11	[[1,9,3],[2,5,6],[4,10,8,7,11]]	[3,3,5]	[15]
12	[[1,9,3],[2,5,6],[4,10,12],[7,11,8]]	[3,3,3,3]	[3]
13	[[1,11,4,12,10,9,3],[2,6],[5,7,13],[8]]	[7,2,3,1]	[42]
14	[[1,11,9,3],[2,6],[4,12],[5,7,13,10,14],[8]]	[4,2,2,5,1]	[20]
15	[[1,11,9,3],[2,6],[4,12],[5,7,13,15],[8],[10,14]]	[4,2,2,4,1,2]	[4]
16	[[1,13,11,10,16,6,2,7,15,12,4,14,5,8,9,3]]	[16]	[16]
17	[[1,13,17,6,2,7,15,5,8,9,3],[4,14,11,10,16,12]]	[11,6]	[66]
18	[[1,13,17,12,4,14,11,10,16,18,6,2,7,15,5,8,9,3]]	[18]	[18]
19	[[1,15,5,9,3],[2,8,10,18,14,12,4,16,13,19,7,17,6],[11]]	[5,13,1]	[65]
20	[[1,15,5,9,3],[2,8,10,18,6],[4,16,20,7,17,13,19,14,12],[11]]	[5,5,9,1]	[45]
21	[[1,15,5,9,3],[2,8,10,18,6],[4,16,20,14,12],[7,17,13,19,21],[11]]	[5,5,5,5,1]	[5]
22	[[1,17,14,13,21,16,22,8,11,12,4,18,6,2,9,3],[5,10,20,7,19,15]]	[16,6]	[48]
23	[[1,17,14,13,21,7,19,23,8,11,12,4,18,6,2,9,3],[5,10,20,15],[16,22]]	[17,4,2]	[68]
24	[[1,17,14,13,21,7,19,23,16,22,24,8,11,12,4,18,6,2,9,3],[5,10,20,15]]	[20,4]	[20]
25	[[1,19,25,9,3],[2,10,22,17,15,5,11,13,23,8,12,4,20,16,24,18,6],[7,21],[14]]	[5,17,2,1]	[170]
26	[[1,19,25,18,6,2,10,22,26,9,3],[4,20,16,24,8,12],[5,11,13,23,17,15],[7,21],[14]]	[11,6,6,2,1]	[66]
27	[[1,19,25,27,9,3],[2,10,22,26,18,6],[4,20,16,24,8,12],[5,11,13,23,17,15],[7,21],[14]]	[6,6,6,6,2,1]	[6]
28	[[1,21,7,23,18,6,2,11,14,15,5,12,4,22,28,10,24,8,13,25,19,27,20,17,16,26,9,3]]	[28]	[28]
29	[[1,21,7,23,18,6,2,11,14,15,5,12,4,22,28,20,17,16,26,19,27,9,3],[8,13,25,29,10,24]]	[23,6]	[138]
30	[[1,21,7,23,18,6,2,11,14,15,5,12,4,22,28,30,10,24,8,13,25,29,20,17,16,26,19,27,9,3]]	[30]	[30]

表 2: 枚数とサイクル分解 その 2(31〜120 枚)

枚数	サイクル積の型	サイクルの LCM	軸
31	[28,1,2]	[28]	17
32	[31,1]	[31]	17
33	[16,16,1]	[16]	17
34	[30,4]	[60]	
35	[35]	[35]	
36	[18,18]	[18]	
37	[4,4,9,2,4,4,2,4,3,1]	[36]	20
38	[4,4,4,2,4,4,2,4,7,1,2]	[28]	20
39	[4,4,4,2,4,4,2,4,4,1,4,2]	[4]	20
40	[40]	[40]	
41	[26,15]	[390]	
42	[42]	[42]	
43	[17,11,10,4,1]	[3740]	23
44	[11,11,21,1]	[231]	23
45	[11,11,11,11,1]	[11]	23
46	[16,22,6,2]	[528]	
47	[25,16,6]	[1200]	
48	[42,6]	[42]	
49	[6,3,3,6,11,3,3,2,3,5,1,3]	[330]	26
50	[6,3,3,6,6,3,3,2,3,3,8,1,3]	[24]	26
51	[6,3,3,6,6,3,3,2,3,3,6,1,3,3]	[6]	26
52	[43,5,4]	[860]	
53	[39,5,5,4]	[780]	
54	[20,20,5,5,4]	[20]	
55	[43,11,1]	[473]	29
56	[55,1]	[55]	29
57	[28,28,1]	[28]	29
58	[9,29,10,10]	[2610]	
59	[19,10,10,10,10]	[190]	
60	[10,10,10,10,10,10]	[10]	
61	[27,8,4,15,2,2,1,2]	[1080]	32
62	[16,8,4,23,2,2,4,1,2]	[368]	32
63	[16,8,4,16,2,8,2,4,1,2]	[16]	32
64	[64]	[64]	
65	[43,22]	[946]	
66	[22,22,22]	[22]	
67	[12,29,4,6,4,11,1]	[3828]	35
68	[12,12,23,6,4,6,4,1]	[276]	35
69	[12,12,12,6,4,6,4,12,1]	[12]	35
70	[11,35,12,12]	[4620]	

枚数	サイクル積の型	サイクルの LCM	軸
71	[23,12,12,12,12]	[276]	
72	[12,12,12,12,12,12]	[12]	
73	[52,18,2,1]	[468]	38
74	[35,18,18,2,1]	[630]	38
75	[18,18,18,18,2,1]	[18]	38
76	[76]	[76]	
77	[74,3]	[222]	
78	[78]	[78]	
79	[7,23,8,8,8,8,8,8,1]	[1288]	41
80	[15,8,8,8,8,8,8,8,8,1]	[120]	41
81	[8,8,8,8,8,8,8,8,8,8,1]	[8]	41
82	[47,16,15,4]	[11280]	
83	[16,16,31,16,4]	[496]	
84	[16,16,16,16,16,4]	[16]	
85	[10,10,5,10,9,5,2,24,5,2,1,2]	[360]	44
86	[10,10,5,10,19,5,2,5,10,2,5,1,2]	[190]	44
87	[10,10,5,10,10,5,2,10,5,10,2,5,1,2]	[10]	44
88	[6,6,6,6,3,6,5,6,3,6,6,17,6,3,3]	[510]	
89	[6,6,6,6,3,6,11,6,3,6,6,6,3,6,6,3]	[66]	
90	[6,6,6,6,3,6,6,6,3,6,6,6,6,3,6,6,3]	[6]	
91	[39,23,22,1,6]	[19734]	47
92	[23,23,45,1]	[1035]	47
93	[23,23,23,23,1]	[23]	47
94	[89,5]	[445]	
95	[42,48,5]	[1680]	
96	[48,48]	[48]	
97	[20,43,4,4,19,4,2,1]	[16340]	50
98	[20,39,20,4,4,4,2,1,4]	[780]	50
99	[20,20,20,4,4,20,4,2,1,4]	[20]	50
100	[33,36,31]	[12276]	
101	[67,34]	[2278]	
102	[34,34,34]	[34]	
103	[99,1,3]	[99]	53
104	[103,1]	[103]	53
105	[52,52,1]	[52]	53
106	[79,27]	[2133]	
107	[27,27,53]	[1431]	
108	[27,27,27,27]	[27]	
109	[12,6,23,12,4,6,6,2,6,6,6,2,11,4,1,2]	[3036]	56
110	[12,6,6,12,4,6,6,12,2,6,6,6,2,17,4,1,2]	[204]	56
111	[12,6,6,12,4,6,6,12,2,6,6,6,2,12,4,1,6,2]	[12]	56
112	[44,11,36,4,17]	[6732]	
113	[44,11,37,4,11,6]	[4884]	
114	[44,11,44,4,11]	[44]	
115	[29,85,1]	[2465]	59
116	[29,29,57,1]	[1653]	59
117	[29,29,29,29,1]	[29]	59
118	[5,5,5,5,5,5,5,5,14,5,5,5,5,5,5,5,5,5,5,5,4,5]	[140]	
119	[5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,9,5,5]	[45]	
120	[5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5]	[5]	

4 考察

結果の章の 2 つのグラフを比較することでわかる事を列挙する.

1. 枚数 N での周期 l は, その枚数でサイクル分解した各サイクルの長さの最小公倍数と一致する.
2. 逆転は, 枚数が 3 の倍数の時にしか発生しない.(但し,4 は除く)
3. 逆転を経て元に戻った周期 l が枚数と一致するとき, その 2 枚少ない時でも周期と枚数が一致している.
4. ただ, 枚数と周期が一致するのは 3 の倍数とは限らない. また, 偶奇とも関係がない.
5. サイクル分解後のサイクルが 1 つの時, 以下の 2 つの何れかが発生する.
   - 周期と枚数が一致する
   - 周期が ((枚数) − 1) 回になる
6. サイクル分解後のサイクルが 1 つで, 且つその要素が 3 の倍数だった時,
   逆転, 周期と枚数の一致が発生する.
7. 枚数 N について,$N-1=3q+r$ と表す.(但し,$0 \leqq r < 3$)
   この時 $q=2k(k=1,2,3,\ldots)$ ならば恒等置換になる数が 1 つ現れ, その値は $3k+2$ である.
   (恒等置換が現れる数 N は, この限りでは無い)

これらついて考察していく.

4.1 サイクル分解について

(予想 1)
枚数 N での周期 l は, その枚数でサイクル分解した各サイクルの長さの最小公倍数と一致する.

考察に入る前に, 実例を 1 つ見る. 使う列は $L_0 = [1, 2, \cdots, 10]$ である.
「サイクル積での配列変化」の 0 回目の部分は,
L_0 と,L_1 とのサイクル分解である
同様に,k 回目の欄は,L_k と L_{k+1} とのサイクル分解である

(例)$N = 10$ の時

表 3: 2 つの場合での配列変化の様子

混ぜた回数	シャッフルでの配列変化	サイクル積での配列変化	分解した回数
0	**[1,2,3,4,5,6,7,8,9,10]**	[**[1,9,8,3]**,**[2,5,6]**,**[4,10]**,[7]]	0
1	[9,5,1,10,6,2,7,3,8,4]	[[9,8,3,1],[5,6,2],[10,4],[7]]	1
2	[8,6,9,4,2,5,7,1,3,10]	[[8,3,1,9],[6,2,5],**[4,10]**,[7]]	2
3	[3,2,8,10,5,6,7,9,1,4]	[[3,1,9,8],**[2,5,6]**,[10,4],[7]]	3
4	[1,5,3,4,6,2,7,8,9,10]	[**[1,9,8,3]**,[5,6,2],**[4,10]**,[7]]	4
5	[9,6,1,10,2,5,7,3,8,4]	[[9,8,3,1],[5,6,2],[10,4],[7]]	5
6	[8,2,9,4,5,6,7,1,3,10]	[[8,3,1,9],**[2,5,6]**,**[4,10]**,[7]]	6
7	[3,5,8,10,6,2,7,9,1,4]	[[3,1,9,8],[5,6,2],[10,4],[7]]	7
8	[1,6,3,4,2,5,7,8,9,10]	[**[1,9,8,3]**,[6,2,5],**[4,10]**,[7]]	8
9	[9,2,1,10,5,6,7,3,8,4]	[[9,8,3,1],**[2,5,6]**,[10,4],[7]]	9
10	[8,5,9,4,6,2,7,1,3,10]	[[8,3,1,9],[5,6,2],**[4,10]**,[7]]	10
11	[3,6,8,10,2,5,7,9,1,4]	[[3,1,9,8],[6,2,5],[10,4],[7]]	11
12	**[1,2,3,4,5,6,7,8,9,10]**	[**[1,9,8,3]**,**[2,5,6]**,**[4,10]**,**[7]**]	12
—	[9,5,1,10,6,2,7,3,8,4]	—	—

この表の右欄の太字になっているのは, 各サイクルで配列が元に戻った事を明示するものである.
この例からも, 各サイクルの長さの最小公倍数がシャッフルの周期と一致することが見て取れる.

また, 長さが r のサイクルの位数は r である事も見て取れる.
この事を示す.

命題

長さ r のサイクルの位数は r

(証明)

長さ r のサイクルを,$\sigma = (a_1, a_2, \cdots, a_r)$ とする

$\sigma^r(a_1) = \sigma^{r-1}(\sigma(a_1)) = \sigma^{r-1}(a_2) = \sigma^{r-2}(\sigma(a_2)) = \sigma^{r-2}(a_3) = \cdots = \sigma(a_r) = a_1$

$\sigma^r(a_2) = \sigma^{r-1}(\sigma(a_2)) = \sigma^{r-1}(a_3) = \sigma^{r-2}(\sigma(a_3)) = \sigma^{r-2}(a_4) = \cdots = \sigma(a_1) = a_2$

$\vdots$

一般に $a_k (1 \leq k \leq n)$ の時

$\sigma^r(a_k) = \sigma^{r-1}(a_{k+1}) = \sigma^{r-2}(a_{k+2}) = \cdots = \sigma^{r-p}(a_{k+p})$

長さは r より,$k + p = r (1 \leqq p \leqq r)$ となった時は ,

$\sigma^{r-p-1}(\sigma(a_{k+p})) = \sigma^{(k+p)-p-1}(\sigma(a_r)) = \sigma^{k-1}(a_1)$ になる.

更に繰り返すと,

$\sigma^{k-2}(a_2) = \sigma^{k-3}(a_3) = \cdots = \sigma(a_{k-1}) = a_k$

よって, 長さ r のサイクルは, いずれの要素も r で元の位置へ戻る.

逆に, 元の位置に戻るための最小位数が r なのは, 上の操作から明らかである.

従って, 長さ r のサイクルの位数は r である.

■

続いて, 次の命題を示す.

命題

3 分シャフリングの周期を m, サイクル積を $\sigma = \tau_1 \tau_2 \cdots \tau_t$, 各サイクルの長さを r_t とする.

この時, 周期 m は次の式で表せる.

$m = \mathrm{lcm}(r_1, r_2, \cdots, r_t)$

(証明)

$\sigma^m = (\tau_1 \tau_2 \cdots \tau_t)^m$

$= (\tau_1 \tau_2 \cdots \tau_t)(\tau_1 \tau_2 \cdots \tau_t) \cdots (\tau_1 \tau_2 \cdots \tau_t)$

互いに素なサイクル積は可換であるため,

$= \tau_1^m \tau_2^m \cdots \tau_t^m$

σが恒等置換になるためには, それぞれのサイクルが一斉に恒等になればいい.

そのためには,

$\tau_1^m = \tau_2^m = \cdots = \tau_t^m = \varepsilon$ ‥ ①

となればよい.

一方, 各サイクルの長さは r_t としたので, 前の命題から

$\tau_1^{r_1} = \tau_2^{r_2} = \cdots = \tau_t^{r_t} = \varepsilon$ ‥ ②

よって,$m = \mathrm{lcm}(r_1, r_2, \cdots, r_t)$ とすれば, ①, ②を満たす.

■

ここで証明した 2 つの命題により, 予想 1 が示された.
これにより, 大きな枚数での 3 分シャフリングの周期 l を手軽に知る事が出来るようになった.

(例)250 枚の場合

```
 ?- nf31(250,50000).
ERROR: Out of global stack
 ?- cycle(250).
250 [55,163,11,5,5,11]-8965
true.
```

実際に 9000 回近く 250 枚のシャッフルを実行している上の例では,
現在の PC では計算が難しくオーバーフローしてしまった.

一方, 下の例では実際にシャッフルしてるのは 1 度で,
あとは サイクル分解→長さを出す→最小公倍数を出すという事をしているだけなので,
より多い枚数の周期を，素早く出す事ができる.
（表 1 で, 周期の値に※が付いていた物は, この方法を使って出した値である.）
〜・〜
また, 予想 1 が片付いた事で, 予想 5 の 1 つが示された事になる.

(予想 5) サイクル分解後のサイクルが 1 つの時, 以下の 2 つの何れかが発生する. ●周期と枚数が一致する ●周期が ((枚数) − 1) 回になる

サイクル積が 1 つの時,
最小公倍数はその数自身になるのは, これまでの議論から明らかである.
もう 1 つの方は, 次の節で述べる.

4.2 「軸」について

章の冒頭の 7 番を見る.

予想 7
枚数 N について,$N-1=3q+r$ と表す.(但し,$r=0,1,2$)
この時 $q=2k(k=1,2,3,\ldots)$ ならば軸が 1 つ現れ, その値は $3k+2$ である.
(軸が現れる枚数 N は, この限りでは無い)

実例として, サイクル分解の表のうち, 軸を持つものをいくつか抜粋する.

表 4: 軸のある場合

枚数	サイクル積	LCM	軸
3	[2,1]	[2]	2
5	[4,1]	[4]	4
7	[3,3,1]	[3]	6
8	[7,1]	[7]	5
9	[4,4,1]	[4]	5
10	[4,3,2,1]	[12]	7
13	[7,2,3,1]	[42]	8
14	[4,2,2,5,1]	[20]	8
15	[4,2,2,4,1,2]	[4]	8
19	[5,13,1]	[65]	11
20	[5,5,9,1]	[45]	11
21	[5,5,5,5,1]	[5]	11
25	[5,17,2,1]	[170]	14
26	[11,6,6,2,1]	[66]	14
27	[6,6,6,6,2,1]	[6]	14
31	[28,1,2]	[28]	17
32	[31,1]	[31]	17
33	[16,16,1]	[16]	17
37	[4,4,9,2,4,4,2,4,3,1]	[36]	20
38	[4,4,4,2,4,4,2,4,7,1,2]	[28]	20
39	[4,4,4,2,4,4,2,4,4,1,4,2]	[4]	20
43	[17,11,10,4,1]	[3740]	23

数が少ない場合は, いくつか例外があるが,
13 枚の時 $(13-1=3*4+0,$ 即ち $k=2$ の時) 以降は, 安定し, なおかつ規則的に軸が出てくる.

今回は, その例外の部分（即ち,$N=3,5,7,10$ の時）は除外し, 安定的な部分を考える.

予想 7 の枠内を整理することで, 命題は次のように書きかえられる.

<u>命題</u>
枚数 N が,$N=6k+r$ $(k\in\mathbb{N}, r=1,2,3,(k,r)\neq(1,1))$ と表せるとき, $3k+2$ は軸である.

(証明)
枚数が $N=6k+3(k=1,2,3,\cdots)$ の時を考える.
$L_0=[1,2,3,\cdots,6k+3]$
要素を 3 等分にし, それぞれのリストに前から A, B, C と名付ける.

$$\mathrm{A}=\begin{bmatrix}1 & ,2 & ,3 & ,\dots & ,k & ,k+1 & ,k+2 & ,\dots & ,2k & ,2k+1\end{bmatrix}$$
$$\mathrm{B}=\begin{bmatrix}2k+2 & ,2k+3 & ,2k+4 & ,\dots & ,3k & ,3k+1 & ,3k+2 & ,\dots & ,4k+1 & ,4k+2\end{bmatrix}$$
$$\mathrm{C}=\begin{bmatrix}4k+3 & ,4k+4 & ,4k+5 & ,\dots & ,5k & ,5k+1 & ,5k+2 & ,\dots & ,6k+2 & ,6k+3\end{bmatrix}$$

ここで, 方法の節で述べたように,
A の要素に, 順番に $a_1,a_2,\dots,a_{2k+1}$ と番号付けを行う.
同様に,B の要素に,$b_1,b_2,\dots,b_{2k+1}$,C の要素に $c_1,c_2,\dots,c_{2k+1}$ と番号付けを行う.

3 分シャフリングを行うと
$L_1=[c_1,b_1,a_1,c_2,b_2,a_2,\cdots,c_k,b_k,a_k,c_{k+1},b_{k+1},a_{k+1},\cdots,c_{2k},b_{2k},a_{2k},c_{2k+1},b_{2k+1},a_{2k+1}]$
L_1 の $3k+2$ 番目の要素を調べる.
右辺のリスト内を 3 つずつに区切る.
$[[c_1,b_1,a_1],[c_2,b_2,a_2],\cdots,[c_k,b_k,a_k],[c_{k+1},b_{k+1},a_{k+1}],\cdots,[c_{2k},b_{2k},a_{2k}],[c_{2k+1},b_{2k+1},a_{2k+1}]]$
よって,$3k+2$ 番目の要素は,b_{k+1} である.
B は, 初項 $2k+2$, 公差 1 の等差数列で,$b_i=2k+1+i$ と表せるので,
$b_{k+1}=3k+2$ となり, 確かにこの場合は $3k+2$ が軸になる.

$N=6k+2,6k+1$ の時については,
c_{2k},c_{2k+1} の要素の有無のみが変化するので軸の位置には関係がない.
よって示された.
■

$N=3,5,7,10$ の場合について
・$N=3$ の時
　そもそも 3 枚未満の 3 分シャフリングを想定していないので,
　「$k=0$ の時は $r\neq 1,2$ である」と定めれば, この場合でも式を満たしている.
・$N=7$ の時
　証明した条件下では,c_{2k} と c_{2k+1} は軸に影響の無い場所にあるので, 一定値 $3k+2$ に定まる.
　しかし,$N=7$ の時は,$(3*1+2=)5$ 番目より前に $c_{2k}(=c_2$ より 4 番目になる$)$ がある為,
　軸がずれてしまっている.
・$N=5,10$ の時については, 理由付けができなかった
　「c_{2k} や c_{2k+1} をどかしたら, たまたま発生した」という帰結に留まっている.

参考として, 上式が成り立っていない枚数では,$3k+2$ 番目の要素はどうなるかを見てみる.

$N=6k$ の時を考える.
リストを３分割すると, １つのリストの長さが $2k$ になるため,
シャッフル後のリストを,3 つずつ区切ると
$[[c_1,b_1,a_1],[c_2,b_2,a_2],\cdots,[c_k,b_k,a_k],[c_{k+1},b_{k+1},a_{k+1}],\cdots,[c_{2k-1},b_{2k-1},a_{2k-1}],[c_{2k},b_{2k},a_{2k}]]$
この時,$3k+2$ 番目に来る要素は, 同じく b_{k+1} となるが,
Ｂの要素は, 初項 $2k+1$, 公差 1 の等差数列となる為,$b_j=2k+j$
よって,$b_{k+1}=3k+1$ となり $3k+2$ とは異なっている.
$t=k-1$ と取り直せば,$N=6t+6$ となる.

$N=6t+4,6t+5$ の場合も c_{2k-1},c_{2k} の要素の有無が変化するだけなので同様である.
よって,$N=6k+r(r=0,4,5)$ の時は, 軸になり得ない.
〜・〜
(2 月 3 日・追記)
卒研発表の際, 軸について,
非常に重要な事でありながら, 盲点としていた点をご指摘いただいた.
追記をする必要性が絶対にあると思われたので, 発表後ではあるが追記する.
川崎先生に感謝申し上げます.

軸については,$N=6k+r(k\in\mathbb{N},r=1,2,3)$ であれば,$3k+2$ が軸になることは述べた通りである.
では, 本当に軸は $3k+2$ の１つなのかを示す.

<u>命題</u>
軸は,$N=6k+r(k\in\mathbb{N},r=1,2,3)$ の時の $3k+2$ 以外に存在しない.

(証明)
枚数が $N=6k+3(k=1,2,3,\cdots)$ の時を考える.
$L_0=[1,2,3,\cdots,6k+3]$
要素を３等分にし, それぞれのリストに前から A, B, C と名付ける.

$$\mathrm{A}=\begin{bmatrix} 1 & ,2 & ,3 & ,\ldots & ,k & ,k+1 & ,k+2 & ,\ldots & ,2k & ,2k+1 \end{bmatrix}$$
$$\mathrm{B}=\begin{bmatrix} 2k+2 & ,2k+3 & ,2k+4 & ,\ldots & ,3k & ,3k+1 & ,3k+2 & ,\ldots & ,4k+1 & ,4k+2 \end{bmatrix}$$
$$\mathrm{C}=\begin{bmatrix} 4k+3 & ,4k+4 & ,4k+5 & ,\ldots & ,5k & ,5k+1 & ,5k+2 & ,\ldots & ,6k+2 & ,6k+3 \end{bmatrix}$$

先程と同様に,
A の要素に, 順番に $a_1,a_2,\ldots,a_{2k+1}$,
B の要素に,$b_1,b_2,\ldots,b_{2k+1}$,C の要素に $c_1,c_2,\ldots,c_{2k+1}$ と番号付けを行う.

3分シャフリングを行うと

$L_1 = [c_1, b_1, a_1, c_2, b_2, a_2, \cdots, c_k, b_k, a_k, c_{k+1}, b_{k+1}, a_{k+1}, \cdots, c_{2k}, b_{2k}, a_{2k}, c_{2k+1}, b_{2k+1}, a_{2k+1}]$

右辺のリスト内を3つずつに区切ると,

$[[c_1, b_1, a_1], [c_2, b_2, a_2], \cdots, [c_k, b_k, a_k], [c_{k+1}, b_{k+1}, a_{k+1}], \cdots, [c_{2k}, b_{2k}, a_{2k}], [c_{2k+1}, b_{2k+1}, a_{2k+1}]]$

ここで,

Aの要素の一般項は$a_i = i$, シャッフル後のa_iの位置は,$3i$番目

Bの要素の一般項$b_i = 2k+1+i$, シャッフル後のb_iの位置は,$3i-1$番目

Cの要素の一般項$c_i = 4k+2+i$, シャッフル後のc_iの位置は,$3i-2$番目 $(i \in \mathbb{N}, 1 \leq i \leq 2k+1)$

と表せる

一般項と, シャッフル後の位置が等しくかどうかを調べる.

(1)Aの要素について

$i = 3i$より,$i = 0$となるので不適

(2)Bの要素について

$2k+1+i = 3i-1$

$2i = 2k+2$

$i = k+1$

よって,Bの要素は$k+1$番目で軸となる.

(3)Cの要素について

$4k+2+i = 3i-2$

$2i = 4k+4$

$i = 2k+2$

しかし,$0 \leq i \leq 2k+1$より不適

よって, 命題の条件下での$3k+2$以外の要素は, 軸になり得ない.

■

また, この時予想 5 の下の方が「一部」示された

(予想 5)
サイクル分解後のサイクルが 1 つの時, 以下の 2 つの何れかが発生する.
●周期と枚数が一致する
●周期が ((枚数) − 1) 回になる

以降,”(枚数)”の部分を N,”周期”の部分を l で示す.
今回調べた中で,$l = N - 1$ となっていたのは, 下の 6 例である.

表 5: $l = N - 1$ の例

枚数	周期	サイクル積の型	$N \div 6$ の式	軸
3	2	[2]	$(3 = 6 \times 0 + 3)$	2
5	4	[4]	−	4
8	7	[7]	$8 = 6 \times 1 + 2$	5
32	31	[31]	$32 = 6 \times 5 + 2$	17
37	36	[4,4,9,2,4,4,2,4,3]	$37 = 6 \times 6 + 1$	20
56	55	[55]	$56 = 6 \times 9 + 2$	29

『サイクル積が 1 つのみである』という事だけを前提とすれば,
$N = 3, 5, 8, 32, 56$ の時, 軸として 1 つの数が, サイクル分解の輪からはずれている.

この事から,

枚数Nでのサイクル積が単体だった時, 周期 l は $r \equiv N \pmod 6$ として,
● $r = 1, 2, 3 \Rightarrow l = N - 1$
● $r = 0, 4, 5 \Rightarrow l = N$

ということがいえる.

更に, この式から予想 4 についての考察を加える.

枚数と周期が一致しているもののサイクルをみると,
今回研究の素材とした 120 枚以下の例では, 全て単体のサイクルで表せていた.

よって, 予想 4 の議論は, 前ページの式の議論に集約できる.
これによれば,
・「枚数と周期が一致するのは 3 の倍数とは限らない」
　⇒ 単一サイクルの長さが,6 を法として 4 または 5 と合同の時である.
・「枚数と周期が一致するのは, 偶奇とは関係がない」
　⇒ 単一サイクルの長さが,6 を法として 5 と合同の時である.
と結論付けられる.

(予想 4)
枚数と周期が一致するのは 3 の倍数とは限らない. また, 偶奇とも関係がない.

表 6: 枚数と周期が一致している例

枚数	周期	サイクル積の型	$N \div 6$ の式	逆転の有無
4	4	[4]	$4 = 6 \times 0 + 4$	○
6	6	[6]	$6 = 6 \times 1 + 0$	○
16	16	[16]	$16 = 6 \times 2 + 4$	
18	18	[18]	$18 = 6 \times 3 + 0$	○
28	28	[28]	$28 = 6 \times 4 + 4$	
30	30	[30]	$30 = 6 \times 5 + 0$	○
35	35	[35]	$35 = 6 \times 5 + 5$	
40	40	[40]	$40 = 6 \times 6 + 4$	
42	42	[42]	$42 = 6 \times 7 + 0$	○
64	64	[64]	$64 = 6 \times 10 + 4$	
76	76	[76]	$76 = 6 \times 12 + 4$	
78	78	[78]	$78 = 6 \times 13 + 0$	○

ただ, 肝心の「どういう時にサイクルが単体になり, どういう時に複数になるのか」が, 最後までわからなかった.
また, 前ページの式も, 軸の面や,$l = N - 1$ の面から言って, 例外がかなり多いので, 更に包括的な式が無いかどうか気になるところである.

4.3 逆転について

表 1 から, 逆転が生じているものだけを抜き出し, 改めて提示する.

このように見ると,3 つのパターンに分類ができる.
①サイクル積が単数の時
②異なる長さのサイクルが複数ある時
②' 全て同じ長さのサイクルが複数ある

特徴を見るために, ①と②について, シャッフルとサイクル積の 2 面から変化をみてみる.

表 7: 逆転に関するデータ

枚数	逆転周期	周期	サイクル分解の型	$l = N$ か
3	1	2	[2]	
4	2	4	[4]	○
6	3	6	[6]	○
9	2	4	[4,4]	
18	9	18	[18]	○
24	10	20	[20,4]	
27	3	6	[6,6,6,6,2]	
30	15	30	[30]	○
33	8	16	[16,16]	
36	9	18	[18,18]	
42	21	42	[42]	○
48	21	42	[42,6]	
57	14	28	[28,28]	
60	5	10	[10,10,10,10,10,10]	
66	11	22	[22,22,22]	
72	6	12	[12,12,12,12,12,12]	
75	9	18	[18,18,18,18,2]	
78	39	78	[78]	○
81	4	8	[8,8,8,8,8,8,8,8,8,8]	
96	24	48	[48,48]	
102	17	34	[34,34,34]	
105	26	52	[52,52]	

表 8: パターン①：単数サイクルでの逆転の検証 $N = 18$ の場合

混ぜた回数	シャッフルでの変化	要素の公差	サイクル積での変化
—	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18]	1	[[1,13,17,12,4,14,11,10,16,18,6,2,7,15,5,8,9,3]]
1	[13,7,1,14,8,2,15,9,3,16,10,4,17,11,5,18,12,6]	6	[[13,17,12,4,14,11,10,16,18,6,2,7,15,5,8,9,3,1]]
2	[17,15,13,11,9,7,5,3,1,18,16,14,12,10,8,6,4,2]	2	[[17,12,4,14,11,10,16,18,6,2,7,15,5,8,9,3,1,13]]
3	[12,5,17,10,3,15,8,1,13,6,18,11,4,16,9,2,14,7]	7	[[12,4,14,11,10,16,18,6,2,7,15,5,8,9,3,1,13,17]]
4	[4,8,12,16,1,5,9,13,17,2,6,10,14,18,3,7,11,15]	4	[[4,14,11,10,16,18,6,2,7,15,5,8,9,3,1,13,17,12]]
5	[14,9,4,18,13,8,3,17,12,7,2,16,11,6,1,15,10,5]	5	[[14,11,10,16,18,6,2,7,15,5,8,9,3,1,13,17,12,4]]
6	[11,3,14,6,17,9,1,12,4,15,7,18,10,2,13,5,16,8]	8	[[11,10,16,18,6,2,7,15,5,8,9,3,1,13,17,12,4,14]]
7	[10,1,11,2,12,3,13,4,14,5,15,6,16,7,17,8,18,9]	9	[[10,16,18,6,2,7,15,5,8,9,3,1,13,17,12,4,14,11]]
8	[16,13,10,7,4,1,17,14,11,8,5,2,18,15,12,9,6,3]	3	[[16,18,6,2,7,15,5,8,9,3,1,13,17,12,4,14,11,10]]
9	[18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1]	1	[[18,6,2,7,15,5,8,9,3,1,13,17,12,4,14,11,10,16]]
—	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18]	—	—

表 9: パターン②：複数サイクルでの逆転の検証 $N = 24$ の場合

	シャッフルでの変化	d	サイクル積での配列変化
0	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24]	1	[[1,17,14,13,21,7,19,23,16,22,24,8,11,12,4,18,6,2,9,3],[5,10,20,15]]
1	[17,9,1,18,10,2,19,11,3,20,12,4,21,13,5,22,14,6,23,15,7,24,16,8]	8	[[17,14,13,21,7,19,23,16,22,24,8,11,12,4,18,6,2,9,3,1],[10,20,15,5]]
2	[14,3,17,6,20,9,23,12,1,15,4,18,7,21,10,24,13,2,16,5,19,8,22,11]	11	[[14,13,21,7,19,23,16,22,24,8,11,12,4,18,6,2,9,3,1,17],[20,15,5,10]]
3	[13,1,14,2,15,3,16,4,17,5,18,6,19,7,20,8,21,9,22,10,23,11,24,12]	12	[[13,21,7,19,23,16,22,24,8,11,12,4,18,6,2,9,3,1,17,14],[15,5,10,20]]
4	[21,17,13,9,5,1,22,18,14,10,6,2,23,19,15,11,7,3,24,20,16,12,8,4]	4	[[21,7,19,23,16,22,24,8,11,12,4,18,6,2,9,3,1,17,14,13],[5,10,20,15]]
5	[7,14,21,3,10,17,24,6,13,20,2,9,16,23,5,12,19,1,8,15,22,4,11,18]	7	[[7,19,23,16,22,24,8,11,12,4,18,6,2,9,3,1,17,14,13,21],[10,20,15,5]]
6	[19,13,7,1,20,14,8,2,21,15,9,3,22,16,10,4,23,17,11,5,24,18,12,6]	6	[[19,23,16,22,24,8,11,12,4,18,6,2,9,3,1,17,14,13,21,7],[20,15,5,10]]
7	[23,21,19,17,15,13,11,9,7,5,3,1,24,22,20,18,16,14,12,10,8,6,4,2]	2	[[23,16,22,24,8,11,12,4,18,6,2,9,3,1,17,14,13,21,7,19],[15,5,10,20]]
8	[16,7,23,14,5,21,12,3,19,10,1,17,8,24,15,6,22,13,4,20,11,2,18,9]	9	[[16,22,24,8,11,12,4,18,6,2,9,3,1,17,14,13,21,7,19,23],[5,10,20,15]]
9	[22,19,16,13,10,7,4,1,23,20,17,14,11,8,5,2,24,21,18,15,12,9,6,3]	3	[[22,24,8,11,12,4,18,6,2,9,3,1,17,14,13,21,7,19,23,16],[10,20,15,5]]
10	[24,23,22,21,20,19,18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1]	1	[[24,8,11,12,4,18,6,2,9,3,1,17,14,13,21,7,19,23,16,22],[20,15,5,10]]
—	[8,16,24,7,15,23,6,14,22,5,13,21,4,12,20,3,11,19,2,10,18,1,9,17]	—	—

読み取れるのは, 以下の事である.

・逆転は,4 を除くと枚数 N が 3 の倍数の時にだけ発生している.

・分解したサイクルの要素の長さは, 全て偶数である

・サイクルの部分を半分で分割し, それぞれを加えると全て同じ値になる.

(この操作で求まる値を「サイクル和」と呼ぶこととする.)

(例)$N = 18$ の時

サイクルは,[[1,13,17,12,4,14,11,10,16,18,6,2,7,15,5,8,9,3]] である.

これを半分で分割し, それぞれ加えると

$$
\begin{array}{rrrrrrrrrr}
 & 1 & 13 & 17 & 12 & 4 & 14 & 11 & 10 & 6 \\
+) & 18 & 6 & 2 & 7 & 15 & 5 & 8 & 9 & 3 \\
\hline
 & 19 & 19 & 19 & 19 & 19 & 19 & 19 & 19 & 19
\end{array}
$$

・一回シャッフルするごとに, サイクル内の要素は 1 つ左方にずれていき,

サイクル内の最大の要素が最も左に来ると, そのサイクルの要素は逆転した並び方をする

(「サイクルの長さの半分だけ, サイクルがずれると逆転する.」ともいえるかもしれない)

・シャッフルの配列は, 何度シャッフルをしても

一定の規則を持った等差数列のような配列をしている.

3 つ目についてであるが, このような性質を持つサイクルは他にも存在している.

(例)$N=26$ のサイクル積のうち, 長さが 6 のサイクルの部分

$N=26$ の時のサイクル積は以下の通りである.

[[1,19,25,18,6,2,10,22,26,9,3],[4,20,16,24,8,12],[5,11,13,23,17,15],[7,21],[14]]

いずれも,$4+24=20+8=16+12=28$,$5+23=11+17=13+15=28$ となり,

サイクル和が一致している.(ちなみに,$7+21=28$ であり, 値は一致している)

シャッフルでみてみると,

	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26]
1	[19,10,1,20,11,2,21,12,3,22,13,4,23,14,5,24,15,6,25,16,7,26,17,8,18,9]
2	[25,22,19,16,13,10,7,4,1,26,23,20,17,14,11,8,5,2,18,24,21,9,15,12,6,3]
3	[18,26,25,24,23,22,21,20,19,9,17,16,15,14,13,12,11,10,6,8,7,3,5,4,2,1]

2 つの長さ 6 のサイクルと, 長さ 2 のサイクルが, 全て「半分ずれる」3 回目のシャッフルの時,

かなり逆転に近い様相を示している.

実際, 表で配列として逆転しているものを太字にし, 逆転していないものには下線を引くと,

[<u>18</u>,**26,25,24,23,22,21,20,19**,<u>9</u>,**17,16,15,14,13,12,11,10**,<u>6</u>,**8,7**,<u>3</u>,**5,4,2,1**]

となり, 少なくとも件のサイクル部分の要素は, 全て逆転している.

この時に長さ 11 のサイクルの方も, ほとんど逆転しているが,

結局元の配列に戻るまでに, 順番が完全に逆転することはなかった.

この後, 一般の場合のサイクル積を出す必要性が不可避となった為, 試行錯誤を繰り返したが,

場合分けが厖大になり, 不本意ながら成果を挙げることが出来なかった.

参考として, 数ではなく項の位置としてのサイクル分解を以下に示す.

サイクル和が, 枚数+1 になっている辺りにヒントがありそうな気がする.

文字の付け方は,「方法」の章で述べたものをそのまま採用する.

表 10: 逆転する枚数での, 文字列のサイクル分解

枚数	サイクル積 (「$\Vert$」は中間部分)	サイクル和
3	$[[a_1 \Vert c_1]]$	4
6	$[[a_1, c_1, b_2 \Vert c_2, a_2, b_1]]$	7
9	$[[a_1, c_1 \Vert c_3, a_3], [a_2, b_1 \Vert c_2, b_3]]$	10
18	$[[a_1, c_1, c_5, b_6, a_4, c_2, b_5, b_4, c_4 \Vert c_6, a_6, a_2, b_1, c_3, a_5, b_2, b_3, a_3]]$	19
24	$[[a_a, c_1, b_6, b_5, c_5, a_7, c_3, c_7, b_8, c_6 \Vert c_8, a_8, b_3, b_4, a_4, c_2, a_6, a_2, b_1, a_3], [a_5, b_2 \Vert c_4, b_7]]$	25
27	$[[a_1, c_1, c_7 \Vert c_9, a_9, a_3], [a_2, b_1, c_4 \Vert c_8, b_9, a_6], [a_4, c_2, b_7 \Vert c_6, a_8, b_3], [a_5, b_2, b_4 \Vert c_5, b_8, b_6], [a_7 \Vert c_3]]$	28
30	$[[a_1, c_1, a_7, c_3, b_8, a_6, a_2, b_1, b_4, b_5, a_5, b_2, a_4, c_2, c_8 \Vert c_{10}, a_{10}, c_4, a_8, b_3, c_5, c_9, b_{10}, b_7, b_6, c_6, b_9, c_7, a_9, a_3]]$	31

5　結び

5.1　感想

そもそも, この大学への入学のきっかけになったのが, 高校時代に使っていた数学の教科書について, 「この著者の飯高茂ってどんな人なんだろう」と思ったのが始まりでした.
当時, 大学が「崇高なる最高学府」といったイメージがあり,「きっと, かなり厳格な人に違いない」という先入観を持って, 恐る恐るオープンキャンパスに赴き,
「あれあれ？」という印象を抱いたのと同時に, 「ああ, ここでだったら絶対楽しく数学をやれるな」という確信を抱いたのを今でも思い出します.

浪人して, 入学して所属したルームが, その教科書を共著されてた松本先生のルームだったので,
「じゃあ, 出る時は飯高先生だな」と言う事で飛び込みました.
（といいつつ, 本当はもっと考えましたけれども）

Prolog という, それまで一度も聞いた事が無いプログラムを使っての研究と言う事で,
「大丈夫かな？」という思いはありましたが,
個人的には, それまでに学んだプログラム言語の中では, 一番使いやすかったので苦労を感じた事は正直無いです.
むしろ, 使えば使うほど「器用に出来てるもんだなぁ」という思いが沸いてきて, 研究もその度捗っていった感じがします.

先生から頂いた本を使って, 丸一日不眠でサイクル分解や最小公倍数などのプログラムを組んで,
それが思い通りに動いた感動は何にも代えがたいものでした.
こういった機会を与えて頂いてありがとうございました.

就職先の関係から, 飯高先生とはこれからも長いお付き合いになるかと思いますので,
「これからもよろしくお願いします」と申し上げて, 結びとしたいと思います.

5.2　参考文献

飯高 茂『Prolog で作る数学の世界 for Windows』