Shuffling Cards

Teo Reu, Paweł Narkiewicz, Pavol Kollár
Counsellor: Julia Stadlmann

August 19, 2018

Contents

1 Introduction 2

2 Perfect Two-Handed Riffle Shuffles 3

3 Perfect d-Handed Riffle Shuffles 4

4 Combinations of different Perfect Shuffles 5

5 Arbitrary Riffle Shuffles 6

6 Arbitrary Shuffles 7

7 Composing Different Two-Handed Riffle Shuffles 9

8 Codes 10
8.1 Perfect Two-Handed Shuffle 10
8.2 Perfect Three-Handed Shuffle 10
8.3 Riffle Shuffle 11

1 Introduction

In this project we explore the shuffling of cards. In particular, we are interested in the periods of different shuffles, that is, when we repeat a given shuffle, how long does it take until we return to the original position?

Let us define a bit of terminology:

Definition 1. *If we split the deck of cards into two stacks and mix these together such that the order of cards in respective stacks does not change then we call this a "Two-Handed Riffle Shuffle". For the rest of this document, we will mostly use the abbreviation "THRS".*

Example 1. An example of such a shuffle with ten cards looks like this:

$$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

We split the deck into two piles:

$$(1, 2, 3, 4); (5, 6, 7, 8, 9, 10)$$

And mix them together:

$$(1, 5, 2, 6, 7, 3, 8, 4, 9, 10)$$

Definition 2. *Performing a THRS such that the two stacks are equal and the cards alternate throughout the whole shuffle is a "Perfect Two-Handed Riffle Shuffle". We will mostly use the abbreviation "PTHRS" from now on.*

Example 2. An example of such shuffle with ten cards looks like this:

$$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

We split the deck into two equal piles:

$$(1, 2, 3, 4, 5); (6, 7, 8, 9, 10)$$

And mix them together in alternating manner:

$$(1, 6, 2, 7, 3, 8, 4, 9, 5, 10)$$

Definition 3. *A PTHRS that keeps the top and bottom cards unchanged is called "Out-Shuffle" or in our case "Left Riffle shuffle". We will mostly use its abbreviation "LRS" from now on.*

Example 3. An example of such shuffle with twelve cards could look like this:

$$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$$

We split the deck into two equal piles:

$$(1, 2, 3, 4, 5, 6); (7, 8, 9, 10, 11, 12)$$

And mix them together in alternating manner, such that the top and bottom card do not move:

$$(1, 7, 2, 8, 3, 9, 4, 10, 5, 11, 6, 12)$$

Definition 4. *A PTHRS that does not keep the top and bottom cards in the same place is called "In-Shuffle" or in our case "Right Riffle Shuffle". We will mostly use its abbreviation "RRS" from now on.*

Example 4. An example of such shuffle with twelve cards could look like this:

$$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$$

We split the deck into two equal piles:

$$(1, 2, 3, 4, 5, 6); (7, 8, 9, 10, 11, 12)$$

And mix them together in alternating manner, such that the top and bottom card do move:

$$(7, 1, 8, 2, 9, 3, 10, 4, 11, 5, 12, 6)$$

2 Perfect Two-Handed Riffle Shuffles

In this section we will look at the PTHRS and how to figure out its period.
Consider periods of PTRHS used in some games. There are many card games which are played with 52 cards. We can see that it takes 8 perfect shuffles to bring them back into original order if we do the LRS but 52 if we do the RRS. In some cases there are two jokers, which means 54 cards and here the RRS is more efficient. Other games might involve 104 cards, which do not have particurarly interesting periods, namely 30 for LRS and 39 for RRS, 108 cards which is also nothing too impressive, 28 LRS, 82 RRS. If we perform LRS with a deck of 32 cards, the period is only 5! In each of these cases, the "Out shuffle" or LRS seems to take fewer shuffles, except for the case of 54 cards. As defined in first section, there are two types of PTHRS. Surprisingly to us at first, these two shuffles behave in the same way.

Theorem 1. *A RRS performed on deck of $2n$ cards permutes these $2n$ cards in the same way that a LRS permutes the middle $2n$ cards of deck of size $2n+2$.*

Now let us prove this by looking at what such shuffles look like.

Proof. Let us look at each of the shuffles individually and see what permutation they create. We will label each of the cards by a number from 1 to $2n$ (or with letters A, B in the second case). The starting order of the deck before performing RRS is:

$$(1, 2, 3, \dots, n-2, n-1, n; n+1, n+2, n+3, \dots, 2n-2, 2n-1, 2n)$$

Splitting the deck into two equal stacks:

$$(1, 2, 3, \dots, n-2, n-1, n)$$
$$(n+1, n+2, n+3, \dots, 2n-2, 2n-1, 2n)$$

And then interweaving them:

$$(n+1, 1, n+2, 2, n+3, 3, \dots; \dots, 2n-2, n-2, 2n-1, n-1, 2n, n)$$

With LRS we need two extra cards, let us call them A, B. Starting order of the deck before performing LRS is:

$$(A, 1, 2, 3, \dots, n-2, n-1, n; n+1, n+2, n+3, \dots, 2n-2, 2n-1, 2n, B)$$

Splitting the deck into two equal stacks:

$$(A, 1, 2, 3, \dots, n-2, n-1, n)$$
$$(n+1, n+2, n+3, \dots, 2n-2, 2n-1, 2n, B)$$

And then interweaving them:

$$(A, n+1, 1, n+2, 2, n+3, 3, \dots; \dots, 2n-2, n-2, 2n-1, n-1, 2n, n, B)$$

We can see, the $2n$ cards truly permuted in the same way.

□

Therefore if we want to prove something for a PTHRS, it is sufficient to do the proof for RRS, unless the deck size is 2. However, this is not an interesting case for LRS because it does not do anything with the arrangement of the cards in the deck.
Let us look at a few examples before moving on.

Number Of Cards	0	2	4	6	8	10	12	14	16	18	20	22	24
RRS period	0	2	4	3	6	10	12	4	8	18	6	11	20
LRS period	0	1	2	4	3	6	10	12	4	8	18	6	11

We can see that the powers of 2 have very short periods with LRS. Is this a coincidence? The next theorem will clear this up.

Theorem 2. *The length of the period of a RRS of $2n$ cards is equal to the order of 2 in $\mathbb{Z}_{2n+1}$.*

Proof. Let us consider a deck of $2n$ cards. Each card will be denoted by an element of $\mathbb{Z}_{2n+1}$. Denote the following as the "first arrangement":

$$(1, 2, 3, \dots, n-2, n-1, n; -n, -n+1, -n+2, \dots, -3, -2, -1)$$

Multiplying every number in the deck by 2 gives us following sequence of labels on the cards:

$$(2, 4, 6, \dots, 2n-4, 2n-2, 2n; -2n, -2n+2, -2n+4, \dots, -6, -4, -2)$$

After exactly one PTHRS applied to the first arrangement of the deck, it will look like this "second arrangement":

$$(-n, 1, -n+1, 2, -n+2, 3, \dots; \dots - 3, n-2, -2, n-1, -1, n)$$

We can see that after the PTRHS each card from the "first arrangement" ended on the same place which it has in the sequence in the middle. That means, that a perfect shuffle essentially multiplies all positions of cards by 2. Therefore the period of such shuffle depends only on order of the number 2 in $\mathbb{Z}_{2n+1}$. □

Below, there is a table of the lengths of periods for given sizes of the stack. This table is equivalent to a table of orders of 2 in different bases. The base is "Number Of Cards" + 1 in case of RRS and "Number Of Cards" - 1 in case of LRS.

Number Of Cards	0	2	4	6	8	10	12	14	16	18	20	22	24
RRS period	0	2	4	3	6	10	12	4	8	18	6	11	20
LRS period	0	1	2	4	3	6	10	12	4	8	18	6	11

Number Of Cards	26	28	30	32	34	36	38	40	42	44	46	48	50
RRS period	18	28	5	10	12	36	12	20	14	12	23	21	8
LRS period	20	18	28	5	10	12	36	12	20	14	12	23	21

Number Of Cards	52	54	56	58	60	62	64	66	68	70	72	74	76
RRS period	52	20	18	58	60	6	12	66	22	35	9	20	30
LRS period	8	52	20	18	58	60	6	12	66	22	35	9	20

Number Of Cards	78	80	82	84	86	88	90	92	94	96	98	100	102
RRS period	39	54	82	8	28	11	12	10	36	48	30	100	51
LRS period	30	18	28	5	10	12	36	12	20	14	12	23	21

Number Of Cards	104	106	108	110	112	114	116	118	120	122	124	126	128
RRS period	39	54	82	8	28	11	12	10	36	48	30	100	51
LRS period	30	18	28	5	10	12	36	12	20	14	12	23	21

3 Perfect d-Handed Riffle Shuffles

Let us consider the general case of d stacks of cards. We can find a generalization of the result on periods for THPS - many of the amazing results from earlier still hold for the d handed shuffle.

Theorem 3. *The length of the period of a d-handed shuffle of a deck with dm cards is the order of d in $\mathbb{Z}_{dm-1}$.*

Proof. Consider a deck of cards:

$$0, 1, \dots, dm-1$$

Firstly, notice that 0 and $dm-1$ do not change their positions after one shuffle. 0 is always at the top of the stack and $dm-1$ takes always the last position. The positions of

$$1, 2, \dots, dm-2$$

are much more interesting than 0 and $dm-1$. We will use $\mathbb{Z}_{dm-1}$ to calculate the positions of these cards efficiently.

Definition 5. *Let $f_n(k) : \mathbb{Z}^\star_{dm-1} \to \mathbb{Z}^\star_{dm-1}$ be the function which assigns to each of the cards 1, 2, ..., $dm-2$ its position after n shuffles.*

Based on our previous observations, let us prove that a position of k is multiplied by d after one shuffle. Consider $k \in \mathbb{Z}_{dm-1}$. Divide k by m using the division algorithm:

$$\exists\, q, r \ \in \mathbb{Z}_{dm-1} : \ k = qm + r \text{ and } 0 \leq r < m$$

Then, after splitting the deck into m piles, k is in qth pile (including 0th pile) and takes the rth position from the top (including 0th position). Look at the drawing below.

$$\begin{array}{llllll}
0 & m & \dots & qm & \dots & (d-1)m \\
1 & m+1 & \dots & qm+1 & \dots & (d-1)m+1 \\
2 & m+2 & \dots & qm+2 & \dots & (d-1)m+2 \\
\vdots & \vdots & \dots & \vdots & \dots & \vdots \\
\vdots & \vdots & \dots & qm+r & \dots & \vdots \\
\vdots & \vdots & \dots & \vdots & \dots & \vdots \\
m-1 & 2m-1 & \dots & (q+1)m-1 & \dots & dm-1
\end{array}$$

Next we can see that all numbers which are closer to the top are before k after one shuffle. Moreover, numbers in the same row, but in earlier column are before k as well. Hence, after one shuffle there are $dr + q$ cards before k and the position of k after one shuffle (including 0-th position) is equal to

$$dr + q.$$

On the other hand (using the fact that $dm = 1$ in $\mathbb{Z}_{dm-1}$)

$$kd = (qm + r)d = qmd + rd = q + dr = dr + q$$

and so in $\mathbb{Z}_{dm-1}$ the position is kd. We can observe that

$$f_n(k) = f(f_{n-1}(k)) = d \cdot f_{n-1}(k)$$

Expanding the recursive formula, we get a pretty nice result, which is:

$$f_n(k) = f \dots f_0(k) = d^n \cdot f_0(k) = d^n \cdot k.$$

In particular

$$f_n(k) = f_0(k) = k \text{ iff } d^n = 1$$

Finally, every card will take the starting position after n shuffles iff $d^n = 1$. The lowest such number is Ord(d). Hence the period of a DHRS is Ord(n). □

4 Combinations of different Perfect Shuffles

We already know that the period of a DHRS on a stack of size md is the order of d in $\mathbb{Z}_{dm-1}$. But what can happen if we combine different shuffles? Surprisingly, there is a very nice pattern. Firstly, let us look at a numerical example. Number 12 is a very good number - it is quite composite - $12 = 2^2 \cdot 3$ - and small as well. The shuffles that we can perform are 1-handed, 2-handed, 3-handed, 4-handed, 6-handed, 12-handed. Let us perform a 2-handed shuffle.

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$$
$$1, 7, 2, 8, 3, 9, 4, 10, 5, 11, 6, 12$$

To be honest, nothing amazing happened. Let us perform now 6-handed shuffle on the last stack:

$$1, 7, 2, 8, 3, 9, 4, 10, 5, 11, 6, 12$$
$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$$

Wow! Something magical has happened - the stack has returned to the original one. Unfortunately, it is not a magic. The Lemma below gives an easy explanation.

Lemma 1. *Suppose m is a size of a stack and $m = ab$. Then if we perform a-handed shuffle and then b-haneded shuffle then we get the original stack.*

Proof. Consider the proof of Theorem 3. Take a card in position k. Then afteran a-handed shuffle its position increases a times in $\mathbb{Z}_{\gtrdot-\nVdash}$ and it is equal to ak, but then after a b-handed shuffle is performed, the position increases b times. $(ab)k = 1 \cdot k = k$ in $\mathbb{Z}_{ab-1}$. □

We have now proved that for every divisior of m, there exists an inverse shuffling. If we perform a_1, a_2, …, a_k-handed shuffles such that $m = a_1 a_2 \cdot \ldots \cdot a_k$, then we also get the original stack! This is a direct consequence of the next Lemma.

Lemma 2. *A combination of a-handed shuffle and b-handed shuffle is equivalent to ab-handed shuffle.*

Proof. The proof is very similar to the previous one. A position of k increases a times after a-handed shuffle, then b times after b-handed shuffle. So the final position of k is ab. It's the same as k after ab-handed shuffle. □

Let us do something more interesting with this fact. We know that if $m = p_1 p_2 \cdots p_n$ then if we perform p_1-handed, p_2-handed, $\cdots$, p_n-handed shuffle then we get the original stack. An interesting question is how many combinations of different shuffles are there such that they give us the original stack. Firstly, let us examine the case with two shuffles. Let us name the two shuffles A and B. We can assign p_i to A or B in two ways. Hence, there are 2^n ways in which two perfect riffle shuffles can give us the original stack. Let's count the number of different combinations with m shuffles. We can "assign" every prime p_i either to first shuffle, second shuffle, …, or mth shuffle. Hence there are n^m different shuffles.

5 Arbitrary Riffle Shuffles

Just like previously, we will be interested how long can the period of a certain riffle shuffle be.

Theorem 4. *The period of any shuffle is the lcm of all lengths of the cycles in this shuffle.*

Proof. Let C be a card and k the length of a cycle that it lies on. In other words, C returns to the original position after k shuffles and k is the smallest such number, so that C is in its starting position whenever exactly nk ($n \in \mathbb{N}$) shuffles where performed. But if the period of the shuffle is p, then C is also in starting position after p shuffles and so $k|p$. Hence the length of every cycle divides p and so the lcm of all cycle lengths divides p. Conversely, if we perform l shuffles, where l is the lcm of the lengths of all cycles, we are back in the starting position for all cards. □

How to get the period to be the largest possible? We do not know, it is an open question for now. The other question is, if we are given lengths of all the cycles after one shuffle, can we construct an adequate riffle shuffle? Yes, we can, and we will prove it by a fairly simple construction process. But for clarity we will first demonstrate this by example.

Example 5. Let the number of cards be 13 and the lengths of cycles $1, 2, 2, 3, 5$. Then these are the cycles that the shuffle will have:

1	2	2	3	5
1	2	3	4	5
x	6	7	8	9
x	x	x	10	11
x	x	x	x	12
x	x	x	x	13

In each step we give each cycle that does not have length 0 a card. After each row of cards, we substract 1 from each of the cycles. If any values become 0, then we do not add any additional cards to it. The shuffle can be easily constructed from this table. In this case, the first 5 cards are the first pile, which gets mixed up with the second pile. The positions to which these 5 cards go to are the biggest numbers in their respective columns in the table above. For example, 4 goes to 10, which goes to 8, which in turn goes to 4. Similarly, 5 goes to 13, which goes to 12 and so on.

Theorem 5. *A deck of n cards can be shuffled with any lengths of cycles $c_1, c_2, \ldots, c_k$ as long as:*

$$\sum_{i=1}^{k} c_i = n$$

Proof. Let n be the number of cards in our deck, c_a be a length of a cycle and k be the number of cycles of our shuffle. Without loss of generality, let us assume that $c_1 \leq c_2 \leq c_3 \leq \ldots \leq c_{k-1} \leq c_k$. These cycles together "cover" the whole size of the deck, therefore:

$$\sum_{i=1}^{k} c_i = n$$

The starting order of cards in our deck is:

$$(1, 2, 3, \ldots, k-1, k, k+1, k+2, \ldots, n-1, n)$$

And this is how we are going to split them into two piles:

$$(1, 2, 3, \ldots, k-1, k); (k+1, k+2, \ldots, n-1, n)$$

Each card in the first pile will represent one of the cycles. Namely card a, where $1 \leq a \leq k$, will represent the cycle of length c_a. Therefore, no two cards from the first pile will be in the same cycle. Now we will construct the shuffle by explicitely specifying in which order, starting with the top, we place cards.

1. We take a variable a, that starts at $a = 1$ and increases, to denote the cycle we are working on. We also use j to denote the number of cards we have already assigned to a position. Initially $j = 0$.

2. Do we get a cycle of the length c_a if we put original card a in position $j+1$, effectively closing the cycle of a now?

   (a) Yes. Close the cycle by placing this card at position $j+1$, onto the bottom of the current deck. Increase a by 1 and repeat step number 2.

   (b) No. Put the next $k-a+1$ cards from the second pile onto the bottom of the current deck, increase j by $k-a+1$ and repeat step number 2.

That this construction indeed works can be easily verfied by considering where each card in the final shuffle goes. So we have proven that this surprising theorem holds. □

6 Arbitrary Shuffles

Let us talk about something a little bit different. We will stop considering riffle shuffles, and simply allow any card to go anywhere. We already know that a shuffle can be represented as a function, which assigns to each position of a deck to the position after one shuffle. This function is a bijection $\{1, ..., n\} \rightarrow \{1, ...n\}$, because two cards cannot be at the same time on the same position and every position is taken by exactly one card. How many different shuffles are there? This is an easy question, we can assign the first card to n different position, the second card to $n-1$ positions, ..., so there are $n!$ different arbitrary shuffles. More interestingly, we can represent these shuffles using graphs, by labeling the vertices with the numbers of cards, cycles in the graph represting a cycle in the shuffle. This way we can count the number of shuffles with specific cycle lengths. Let us count the number of different shuffles with a given number of cycles of certain periods.

To understand the way of counting different shuffles better, let us examine an arbitrary shuffle with the deck size $n = 8$

$$\begin{aligned} 1 &\to 2 \\ 2 &\to 3 \\ 3 &\to 4 \\ 4 &\to 1 \\ 5 &\to 6 \\ 6 &\to 5 \\ 7 &\to 8 \\ 8 &\to 7 \end{aligned}$$

We notice these cycles:

$$1 \to 2 \to 3 \to 4 \to 1$$
$$5 \to 6 \to 5$$
$$7 \to 8 \to 7$$

In a graph we would draw 3 cycles, one with length 4 and two with length 2 and label their vertices correspondingly. Let us swap 5 and 8 in the drawing. Then the graph represent a shuffle

$$1 \to 2 \to 3 \to 4 \to 1$$
$$8 \to 6 \to 8$$
$$7 \to 5 \to 7$$

and this shuffle is different from the previous one. But if we replace 1 by 4, 2 by 1, 2 by 2, and 4 by 3, the following shuffle is the same as the first, since the cards go the same position.

$$4 \to 1 \to 2 \to 3 \to 4$$
$$5 \to 6 \to 5$$
$$7 \to 8 \to 7$$

More formally we have a configuration

$$v_1 \to v_2 \to v_3 \to v_4$$
$$v_5 \to v_6$$
$$v_7 \to v_8$$

and a drawing with 8 initially unlabeled vertices $v_1, \ldots, v_8$, arranged 3 cylces, two of length 2 and one of length 4. We have 8 vertices to be filled with some numbers. Firstly, we can fill all vertices in 8! ways. Let us count in how many ways we can label the vertices to give us the first shuffle. It doesn't matter if we fill v_5, v_6 with $(5,6)$ or $(6,5)$. Same for the other cycle of length 2. Analogically, the cycle of the length 4 can be labelled in 4 ways: $(1,2,3,4),(2,3,4,1),(3,4,1,2),(4,1,2,3)$. This sort of counting works for any shuffle. Currently, our result is $\frac{8!}{4 \cdot 2 \cdot 2}$. We can also swap the labels of v_5, v_6 with the labels of v_7, v_8. So we counted each shuffle $4 \cdot 2 \cdot 2 \cdot$ times and have to divide our result by 2 (number of cycles with length 2). The final result is $\frac{8!}{4 \cdot 2 \cdot 2 \cdot 2}$. Surprisingly, there exists a quite elegant formula for the general case.

Theorem 6. *We want to count the different shuffles with m cards which have a_1 cycles of the length b_1, a_2 cycles of the length b_2, …, a_n cycles with the length b_n. Then we obtain:*

$$\frac{m!}{\prod_{k=1}^{n}(b_k)^{a_k}a_k!}$$

Proof. There are m numbers and m vertices. First vertex can be filled in m ways, second in $m-1$ ways and so on. Hence, we can fill in all vertices in $m!$ ways. Consider a_k cycles with the length b_k. Every cycle can be rearanged in b_k ways, we can move every cycle clockwise and get the same cycle. Moreover if we have a_k cycles with the same length it doesn't matter if we swap any two of them. Hence, we can reaarrange a_k cycles with the length of b_k in $a_k!$ ways. Finally, there are

$$\frac{m!}{\prod_{k=1}^{n}(b_k)^{a_k}a_k!}$$

different shuffles. □

We are now able to find the probability of obtaining cycles and periods of given length. We know that there are $m!$ possible shufflings in total. Then using our formula we can count the number of configurations with the required properties and divide it by $m!$ to obtain the probability.

7 Composing Different Two-Handed Riffle Shuffles

Now we finally allow the mixing of different arbitrary shuffles. Let m be an even number of cards. We can cut the deck into two equal stacks and shuffle over and over again and we use different shuffles each time. Can we return cards to their original position with just two or three carefully chosen shuffles, with the shuffles looking random to the observer? This is of course a bit of a vague question - what does random mean? But we will try to explain some of our ideas on this topic anyway.

We build a riffle shuffle from the top by splitting the cards into two equal decks and always choosing either a card from the top of one of those two decks to come next in the shuffle. We record this in a sequence S of 0s and 1s, 0 for choosing a card from left and 1 for choosing a card from the right side (this way we can generate all different riffle shuffles). After the first shuffle we say that 1 goes to position a_1, 2 goes to a_2 position and so on. For a non-trivial shuffle there has to be a smallest i for which a_i does not go to $i-th$ position, i will also be the position of the first 1 in our sequence. We know that after the next shuffle card a_i needs to return to its original position. Because a_i was the first card that we choose from right, we know that this position is $m/2+1$. So $a_{m/2+1}$ needs to go position a_i and reverse.
Now let us take a deck with 12 cards and sequence $S = (0, 1, 1, 1, 1, 0, \ldots)$. We notice that after such a shuffle 2 ends up on sixth position, which means when we repeat the shuffle and split the cards, 2 will be the last card in the first stack. It is impossible to return 2 to its initial position after just one shuffle, because after choosing 1 from the first pack we can either choose the original 7 or the current seventh card (originally 3 or 11, in our case, depending on how we chose the next term in S). Similarly if we instead for example choose $S = (0, 1, 1, 0, \ldots)$. We see that 2 needs to be either the second card or the seventh $(7 = 12/2 + 1)$ after one shuffle. We can also find similar results for other cards. We see that many initial shuffles will not work. We need $(0, 0, \ldots)$ or $(0, 1, 1, 1, 1, 1, 0, \ldots)$.

Of course, if we want both shuffles to be the same, the possible choices of 0s and 1 are even more restrictive and maybe appear less random. Recall that a_i needs to got to $a_{m/2+1}$ and reverse. For that, we need the first block of 0s have length $i-1$ and the first block of 1s have length $m/2-i+1$. So half of S will be formed of two big blocks of 0s and 1s. We also know that the first card from second stack will be i after the first shuffle. After another shuffle we should have card $i+1$ after card i. Because it is not in the first stack, it has to be below i. Inductively we have found that the next $m/2-i$ entries of S are 0. We have consumed all the zeros, because the number of them is m/2, and the remaining entries are 1s. So S will have 4 blocks of 0 and 1. Which means we are dividing the deck into 4 big piles and shuffling them togheter.

8 Codes

8.1 Perfect Two-Handed Shuffle

```
# include <iostream>

using namespace std;

int pack1[100], pack2[100], deck[200],i,j,a,b;

int main(){

int n, shuffles;
cout<<"Number␣of␣cards:␣";
cin>>n;
cout<<"Number␣of␣shuffles:␣";
cin>>shuffles;

a=1; b=1;
for(i=1;i<=n;i++)
        if (i<=n/2 )pack1[a++]=i;
                else pack2[b++]=i;

//the shuffling
int periodOfShuffle=1;
while(shuffles){
        a=1; b=1;
        for(i=1;i<=n;i++)
                if(i%2==1) deck[i]=pack2[b++];
                else deck[i]=pack1[a++];

        //creating the new pack1, pack2 -
    //cutting the deck in two equal packs
        a=1; b=1;
        for(i=1;i<=n;i++)
                if(i<=n/2) pack1[a++]=deck[i];
                        else pack2[b++]=deck[i];


        shuffles--;

        int ok=1; //checking integer
        // checking if the new deck is the same as the initial one
        for (int i=1;i<=n;i++) if(deck[i]!=i) ok=0;

    //if they are the same, stop 'while' and show solution
        if(ok==1) {cout<<periodOfShuffle<<'␣'; break;}

periodOfShuffle++;
}
}
```

8.2 Perfect Three-Handed Shuffle

```
# include <iostream>
using namespace std;
```

```
int pack1[100], pack2[100], pack3[100], deck[300], a, b, c, n,
maximumNumberOfShuffles=1000;
int main(){

cout<<”Number␣of␣cards:␣”;
cin>> n;
a=1; b=1; c=1;

//cutting the deck into three equal packs
for(int i=1;i<=n;i++)
        if(i<=n/3) pack1[a++]=i;
        else if(i<=2*n/3 && i>=n/3) pack2[b++]=i;
        else pack3[c++] =i;

//shuffling
int numberOfShuffles=0;
while(maximumNumberOfShuffles){
        numberOfShuffles++;
        a=1; b=1; c=1;

//creating the new deck by choosing one card from each stack
        for(int i=1;i<=n;i++)
                if(i%3==1) deck[i]=pack1[a++];
                        else if (i%3==2) deck[i]=pack2[b++];
                                else deck[i]=pack3[c++];

// cutting the deck into three packs
a=1;b=1;c=1;
for(int i=1;i<=n;i++)
        if(i<=n/3) pack1[a++]=deck[i];
        else if(i<=2*n/3 && i>=n/3) pack2[b++]=deck[i];
        else pack3[c++] =deck[i];

        maximumNumberOfShuffles--;
        int ok=1;

    //checking if the deck is equal to the initial one
        for(int i=1;i<=n;i++) if(pack[i]!=i) ok=0;
        if(ok) {cout<<numberOfShuffles; break;}
}
        }
}
```

8.3 Riffle Shuffle

```
# include <iostream>
# include<fstream>
using namespace std;

ifstream fin(”f.in”);

int choose[100], pack1[100],pack2[100], deck[100];

int main(){

        int numberCards, numberCardsFirstDeck,shuffles;

        fin>>numberCards;
        fin>>numberCardsFirstDeck;

        int i;
        for(i=1;i<=numberCards;i++) fin>>choose[i];
    // generate random shuffle

// first deck
int a=1,b=1;
for(i=1;i<=numberCards;i++)
if(i<=numberCardsFirstDeck) pack1[a++]=i;
else pack2[b++]=i;

// shuffle

fin>>shuffles;
int number=0;

//creating new deck
while(shuffles){
        //
        a=1;b=1;
        for(i=1;i<=numberCards;i++)
                if(choose[i]==0) deck[i]=pack1[a++];
                else deck[i]=pack2[b++];

//creating new packs
int a=1,b=1;
for(i=1;i<=numberCards;i++)
if(i<=numberCardsFirstDeck) pack1[a++]=deck[i];
else pack2[b++]=deck[i];

int ok=1;
number++;
for(int i=1;i<=numberCards;i++) if(deck[i]!=i) ok=0;
if(ok==1) break;

shuffles--;
}
cout<<numberOfShuffles;
}
```