# Appendix 7A: Card Shuffling and Fermat's Little Theorem

In this appendix we will define order in terms of card-shuffling, give a combinatorial proof of Fermat's Little theorem, and discuss quick calculations of powers mod $n$.

## 7.11. Card Shuffling and orders mod $n$

The cards in a 52 card deck can be arranged in $52! \approx 8 \times 10^{67}$ different orders. Between card games we shuffle the cards to make the order of the cards unpredictable. But what if someone can shuffle "perfectly"? How unpredictable will the order of the cards then be? Let's analyze this by carefully figuring out what happens in a "perfect shuffle". There are several ways of shuffling cards, the most common being the *riffle shuffle*. In a riffle shuffle one splits the deck in two, places the two halves in either hand and then drops the cards, using one's thumbs, in order to more-or-less interlace the cards from the two decks.

One begins with a deck of 52 cards and, to facilitate our discussion, we will call the top card, card 1, the next card down, card 2, etc. If one performs a *perfect riffle shuffle*, one cuts the cards into two 26 card halves, one half with the cards 1 through 26, the other half with the cards 27 through 52. An "out-shuffle" then interlaces the two halves so that the new order of the cards becomes (from the top) cards

$$1, 27, 2, 28, 3, 29, 4, 30, \ldots$$

That is cards $1, 2, 3, \ldots 26$ go to positions $1, 3, 5, \ldots 51$; and cards $27, 28, \ldots 52$ go to positions $2, 4, \ldots 52$, respectively. We can give formulas for each half

$$k \to \begin{cases} 2k-1 & \text{for } 1 \le k \le 26, \\ 2k-52 & \text{for } 27 \le k \le 52, \end{cases}$$

These coalesce into one formula $k \to 2k - 1 \pmod{51}$ for all $k, 1 \le k \le 52$. The top and bottom cards do not move, that is $1 \to 1$ and $52 \to 52$, so we focus on understanding the permutation of the other fifty cards:

Any shuffle induces a *permutation* $\sigma$ on $\{1, \ldots, 52\}$.[6] For the out-shuffle, $\sigma(1) = 1, \sigma(52) = 52$ and

$$\sigma(1+m) \text{ is the least positive residue of } 1 + 2m \pmod{51} \text{ for } 1 \le m \le 50.$$

To determine what happens after two or more out-shuffles, we simply compute the function $\sigma^k(.)$ $(= \underbrace{\sigma(\sigma(\ldots\sigma}_{k \text{ times}}(.))))$. Evidently $\sigma^k(1) = 1, \sigma^k(52) = 52$, and then

$$\sigma^k(1+m) \text{ is the least positive residue of } 1 + 2^k m \pmod{51} \text{ for } 1 \le m \le 50.$$

Now $2^8 \equiv 1 \pmod{51}$, and so $\sigma^8(1+m) \equiv 1 + m \pmod{51}$ for all $m$. Therefore eight perfect out-shuffles returns the deck to its original state – so much for the 52! possible orderings!

Eight more perfect out-shuffles will also return the deck to its original state, a total of 16 perfect out-shuffles, and also 24, or 32, or 40, etc. Indeed any multiple of 8. So we see that the order of 2 (mod 51) is 8, and that $2^r \equiv 1 \pmod{51}$ if and only if $r$ is divisible by 8. This shows, we hope, why the notion of order is interesting and exhibits one of the key results (Lemma 7.1.2) about orders.

**Exercise 7.11.1.**† An "in-shuffle" is the riffle shuffle that interlaces the cards the other way; that is, after one shuffle, the order becomes cards $27, 1, 28, 2, 29, \ldots 52, 26$. Analyze this in an analogous way to above, and determine how many "in-shuffles" it takes to get the cards back into their original order.

**Exercise 7.11.2.**† What happens when one performs riffle shuffles on $n$ card decks, with $n$ even?

**Exercise 7.11.3.**‡ Suppose that the dealer alternates between in-shuffles and out-shuffles. How many such pairs of shuffles does it take to get the riffle cards back into their original order?

Persi Diaconis is a Stanford mathematics professor who left home at the age of fourteen to learn from sleight-of-hand legend Dai Vernon.[7] It is said that Diaconis can shuffle to obtain any permutation of a deck of playing cards. We are interested in the highest possible order of a shuffle. To analyze this question, remember that a shuffle can be re-interpreted as a permutation $\sigma$ on $\{1, \ldots, n\}$ (where $n = 52$ for a usual deck). One way to explicitly write down a permutation is to track the orbit of each number. For example, for the permutation $\sigma$ on 5 elements given by

$$\sigma(1) = 4,\ \sigma(2) = 5,\ \sigma(3) = 1,\ \sigma(4) = 3,\ \sigma(5) = 2,$$

1 gets mapped to 4, which gets mapped to 3, and 3 gets mapped back to 1, whereas 2 gets mapped to 5 and 5 gets mapped back to 2, so we can write

$$\sigma = (1, 4, 3)(2, 5).$$

Each of $(1, 4, 3)$ and $(2, 5)$ is a *cycle*, and cycles cannot be decomposed any further. Any permutation can be decomposed into cycles in a unique way, the analogy of the fundamental theorem of arithmetic, for permutations. What is the order of $\sigma$? Now $\sigma^n = (1, 4, 3)^n (2, 5)^n$, so that $\sigma^n(1) = 1,\ \sigma^n(4) = 4$ and $\sigma^n(3) = 3$ if and only

[6] That is, $\sigma : \{1, \ldots, 52\} \to \{1, \ldots, 52\}$ such that the $\sigma(i)$ are all distinct (and so $\sigma$ has an inverse).
[7] Check out this story, and these larger-than-life characters, on Wikipedia.

if 3 divides $n$, while $\sigma^n(2) = 2$ and $\sigma^n(5) = 5$ if and only if 2 divides $n$. Therefore $\sigma^n$ is the identity if and only if 6 divides $n$, and so $\sigma$ has order 6.

**Exercise 7.11.4.** Suppose that $\sigma$ is a permutation on $\{1, \dots, n\}$, and that $\sigma = C_1 \cdots C_k$ where $C_1, \dots, C_k$ are disjoint cycles.

(a) Show that the order of $\sigma$ equals the least common multiple of the lengths of the cycles $C_j$, $1 \le j \le k$.

(b) Use this to find the order of the permutation corresponding to an out-shuffle.

(c) Prove that if $n_1, \dots, n_k$ are any set of positive integers for which $n_1 + \cdots + n_k = n$, then there exists a permutation $\sigma = C_1 \cdots C_k$ on $\{1, \dots, n\}$, where each $C_j$ has length $n_j$.

(d) Deduce that the maximum order, $m(n)$, of a permutation $\sigma$ on $\{1, \dots, n\}$ is given by

$$\max \operatorname{lcm}[n_1, \dots, n_k] \text{ over all integers } n_1, \dots, n_k \ge 1 \text{ for which } n_1 + \cdots + n_k = n.$$

Our goal is to determine $m(52)$, the highest order of any shuffle that Diaconis can perform on a regular deck of 52 playing cards. However it is unclear how to determine $m(n)$ systematically. Working through the possibilities for small $n$, using exercise 7.11.4, we find that

$$\begin{array}{llllll}
m(5) = 6 & \text{obtained from} & 6 = 3 \cdot 2 & \text{and} & 5 = 3 + 2 \\
m(6) = 6 & \text{obtained from} & 6 = 3 \cdot 2 \cdot 1 & \text{and} & 6 = 3 + 2 + 1 \\
m(7) = 12 & \text{obtained from} & 12 = 4 \cdot 3 & \text{and} & 7 = 4 + 3 \\
m(8) = 12 & \text{obtained from} & 12 = 4 \cdot 3 \cdot 1 & \text{and} & 8 = 4 + 3 + 1 \\
m(9) = 20 & \text{obtained from} & 20 = 5 \cdot 4 & \text{and} & 9 = 5 + 4 \\
m(10) = 30 & \text{obtained from} & 30 = 5 \cdot 3 \cdot 2 & \text{and} & 10 = 5 + 3 + 2 \\
m(11) = 30 & \text{obtained from} & 30 = 6 \cdot 5 & \text{and} & 11 = 6 + 5 \\
m(12) = 60 & \text{obtained from} & 60 = 5 \cdot 4 \cdot 3 & \text{and} & 12 = 5 + 4 + 3.
\end{array}$$

No obvious pattern jumps out (at least to the author) from this data, though one observes one technical issue:

**Exercise 7.11.5.**† Show that there is a permutation $\sigma = C_1 \cdots C_k$ on $\{1, \dots, n\}$ of order $m(n)$ in which the length of each cycle is either 1 or a power of a distinct prime.

**Exercise 7.11.6.**† Use the previous exercise to determine $m(52)$.

**Exercise 7.11.7.**‡ Use exercise 5.4.3 to prove that $\log m(n) \sim \sqrt{n \log n}$.

## 7.12. The "necklace proof" of Fermat's Little Theorem

Little Sophie has a necklace making kit, which comes with wires that each accommodate $p$ beads and unlimited supplies of beads of $a$ different colours. How many genuinely *different* necklaces can be Sophie make? Two necklaces are *equivalent* if they can be obtained from each other by a rotation, otherwise they are different; and so Sophie is asking for the number of equivalence classes of sequences of length $p$ where each entry is selected from $a$ possible colours.

Suppose we have a necklace with the $j$th bead having colour $c(j)$ for $1 \le j \le p$. One can rotate the necklace in $p$ different ways: If we rotate the necklace $k$ places for some $k$ in the range $0 \le k \le p-1$, then the $j$th bead will have colour $c(j+k)$ for $1 \le j \le p$, where $c(.)$ is taken to be a function of period $p$. If two of these equivalent necklaces are identical then $c(j+k) = c(j+\ell)$ for all $j$, for some $0 \le k < \ell \le p-1$. Then $c(n+d) = c(n)$ for all $n$, where $d = \ell - k \in [1, p-1]$, and so $c(md) = c(0)$ for all $m$; that is, all of the beads in the necklace have the same colour.

Therefore we have proved that, other than the $a$ necklaces made of beads of the same colour which each belong to an equivalence class of size 1, all other necklaces belong to equivalence classes of size $p$. Since there are $a^p$ possible sequences of length $p$ with $a$ possible colours for each entry, and $a$ sequences that all have the same colour, the total number of equivalence classes (different necklaces) is

$$a + \frac{a^p - a}{p}.$$

In particular, we have established that $p$ divides $a^p - a$ for all $a$, as desired.[8]

**Exercise 7.12.1.** Let $p$ be prime. Let $X$ denote a finite set and $f : X \to X$ where $f^p = i$, the identity map. (Here $f^p$ means composing $f$ with itself $p$ times.) Let $X_{\text{fixed}} := \{x \in X : f(x) = x\}$.

(a) Prove that $|X| \equiv |X_{\text{fixed}}| \pmod p$.
Let $G$ be a finite multiplicative group and $X = \{(x_1, \dots, x_p) \in G^p : \ x_1 \cdots x_p = 1\}$.

(b)† Deduce that $\#\{g \in G : g \text{ has order } p\} \equiv |G|^{p-1} - 1 \pmod p$.

(c) Deduce that if $p$ divides the order of finite group $G$ then $G$ contains an element of order $p$. Combined with Lagrange's Theorem 7.23.1 of appendix 7D, this is an "if and only if" criterion.

**Exercise 7.12.2.** Let $p$ be a given prime.

(a) Use (4.12.3) of appendix 4C to determine the number of irreducible polynomials mod $p$ of prime degree $q$.

(b) Deduce that $q^p \equiv q \pmod p$ for every prime $q$

(c) Deduce Fermat's Little Theorem.

## More combinatorics and number theory

[1] Melvin Hausner, *Applications of a simple counting technique*, Amer. Math. Monthly **90** (1983), 127–129.

## 7.13. Taking powers efficiently

How can one raise a residue class mod $m$ to the $n$th power "quickly", when $n$ is very large? In 1785 Legendre computed high powers mod $p$ by *fast exponentiation*: To determine $5^{65} \pmod{161}$, we write 65 in base 2, that is $65 = 2^6 + 2^1$, so that $5^{65} = 5^{2^6} \cdot 5^{2^1}$. Let $f_0 = 5$ and $f_1 \equiv f_0^2 \equiv 5^2 \equiv 25 \pmod{161}$. Next let $f_2 \equiv f_0^4 \equiv f_1^2 \equiv 25^2 \equiv 142 \pmod{161}$, and then $f_3 \equiv f_0^8 \equiv f_2^2 \equiv 142^2 \equiv 39 \pmod{161}$. We continue computing $f_k \equiv f_0^{2^k} \equiv f_{k-1}^2 \pmod{161}$ by successive squaring: $f_4 \equiv 72$, $f_5 \equiv 32$, $f_6 \equiv 58 \pmod{161}$ and so $5^{65} = 5^{64+1} \equiv f_6 \cdot f_0 \equiv 58 \cdot 5 \equiv 129 \pmod{161}$. We have determined the value of $5^{65} \pmod{161}$ in seven multiplications mod 161, as opposed to 64 multiplications by the more obvious algorithm.

In general to compute $a^n \pmod m$ quickly: Define

$$f_0 = a \text{ and then } f_j \equiv f_{j-1}^2 \pmod m \text{ for } j = 1, 2, \dots, j_1,$$

where $j_1$ is the largest integer for which $2^{j_1} \le n$. Writing $n$ in binary, say as $n = 2^{j_1} + 2^{j_2} + \cdots + 2^{j_\ell}$ with $j_1 > j_2 > \cdots > j_\ell \ge 0$, let $g_1 = f_{j_1}$ and then

[8] We've seen that Fermat's Little Theorem arises in many different contexts. Even its earliest discoverers got there for different reasons: Fermat, Euler and Lagrange were led to Fermat's Little Theorem by the search for perfect numbers, whereas Gauss was led to it by studying the periods in the decimal expansion of fractions (as in section 7.8). It seems to be a universal truth, rather than simply an ad hoc discovery.

$g_i \equiv g_{i-1} f_{j_i} \pmod m$ for $i = 2, 3, \ldots, \ell$. Therefore

$$g_\ell \equiv f_{j_1} \cdot f_{j_2} \cdots f_{j_\ell} \equiv a^{2^{j_1}+2^{j_2}+\cdots+2^{j_\ell}} = a^n \pmod m.$$

This involves $j_\ell + \ell - 1 \le 2j_\ell \le \frac{2 \log n}{\log 2}$ multiplications mod $m$ as opposed to $n$ multiplications mod $m$ by the more obvious algorithm.

One can often use fewer multiplications. For example, for $31 = 1+2+4+8+16$ the above uses 8 multiplications, but we can use just 7 multiplications if, instead, we determine $a^{31} \pmod m$ by computing $a^2 \equiv a \cdot a$; $a^3 \equiv a^2 \cdot a$; $a^6 \equiv a^3 \cdot a^3$; $a^{12} \equiv a^6 \cdot a^6$; $a^{24} \equiv a^{12} \cdot a^{12}$; $a^{30} \equiv a^{24} \cdot a^6 \pmod m$ and finally $a^{31} \equiv a^{30} \cdot a \pmod m$.

These exponents form an *addition chain*, a sequence of integers $e_1 = 1 < e_2 < \ldots < e_k$ where, for all $k > 1$, we have $e_k = e_i + e_j$ for some $i, j \in \{1, \ldots, k-1\}$. In the example above, the binary digits of 31 led to the addition chain $1, 2, 3, 4, 7, 8, 15, 16, 31$, but the addition chain $1, 2, 3, 6, 12, 24, 30, 31$ is shorter.

For most exponents $n$, there is an addition chain which is substantially shorter than $j_\ell + \ell - 1$, though never less than half that size. There are many open questions about addition chains. The best known is Scholz's conjecture that the shortest addition chain for $2^n - 1$ has length $\le n - 1$ plus the length of the shortest addition chain for $n$. For much more on addition chains, see Knuth's classic book [Kn].

## 7.14. Running time: The desirability of polynomial time algorithms

In this section we discuss how to measure how fast an algorithm is. The inputs into the algorithm in the previous section for calculating $a^n \pmod m$ are the integers $a$ and $m$, with $1 \le a \le m$, and the exponent $n$. We will suppose that $m$ has $d$ digits (so that $d$ is proportional to $\log m$). The usual algorithms for adding and subtracting integers with $d$ digits take about $2d$ steps, whereas the usual algorithm for multiplication takes about $d^2$ steps.[9]

**Exercise 7.14.1.** Justify that multiplying two residues mod $m$ together and reducing mod $m$ takes no more than $2d^2$ steps.

The algorithm described in the previous section involves about $c \log n$ multiplications of two residues mod $m$, for some constant $c > 0$, and so the total number of steps is proportional to

$$(\log m)^2 \log n.$$

Is this good? Given any mathematical problem, the cost (measured by the number of steps) of an algorithm to resolve the question must include the time taken to read the input data, which can be measured by the number of digits, $D$, in the input. In this case the input is the numbers $a, m$ and $n$, so that $D$ is proportional to $\log m + \log n$. Now if $a$ and $m$ are fixed and we allow $n$ to grow then the algorithm takes $CD$ steps for some constant $C > 0$, which is $C$ times as long as it takes to read the input. You cannot hope to do much better than that. On the other hand, if $m$ and $n$ are roughly the same size, then the algorithm takes time proportional to $D^3$. We still regard this as fast – any algorithm whose speed is bounded by a polynomial in $D$ is a *polynomial time algorithm* and is considered to be pretty fast.

[9] Since we have to multiply each pair of digits together, one from each of the given numbers.

It is important to distinguish between a mathematical problem and an algorithm used for resolving it. There can be many choices of algorithm and one wants a fast one. However, we might only know a slowish algorithm which, even though it may seem clever, does not necessarily mean that there is no fast algorithm.

Let `P` be the class of problems that can be resolved by an algorithm that runs in polynomial time. Few mathematical problems which belong to `P` and the key question is whether we can identify which problems. We'll discuss `P` in section 10.4.

**Exercise 7.14.2.** Prove that the Euclidean algorithm works in polynomial time.