From "Final 3" to "Final n"

Popular Card Trick Explained and Generalized for Any Size Deck

AUGUSTINE BERTAGNOLLI
Auburn University Montgomery

December 15, 2012

Abstract

While the origins of the "Final 3" card trick are unknown, the trick itself is quite popular, with one (out of several) online tutorials gaining over 1.5 million views[1] on YouTube. We explore the mathematical foundations of the trick as well as derive a generalized formula which will allow us to perform the trick with any size deck.

1 Introduction

Commonly referred to as "The Final 3" card trick, the trick is performed as follows.

1. The magician has a volunteer choose 3 cards out of a well-shuffled deck of 52 and the volunteer is asked to memorize those 3 cards. The volunteer holds the cards without revealing them to the magician.

2. From the remaining 49 cards, the magician makes 3 stacks of face-down cards of sizes (from left to right) 14, 15, and 15, leaving 5 cards in his hand.[2]

3. The volunteer
   - places one of his 3 cards on top of the leftmost pile,
   - takes any number of cards from the middle stack and places them on top of the left stack,
   - places his second card onto what is left of the middle stack,
   - takes any number of cards from the rightmost stack and places them on top of the middle stack,
   - and places his last card onto what remains of the right stack.

4. The magician
   - stacks the cards back into one pile in the order: 5 left in his hand onto the rightmost stack, all this onto the middle stack, and all this onto the left stack,
   - asks the volunteer to let him know if he sees any one of his 3 cards as he begins flipping from the top of the deck in the order *face up, face down, face up, face down, etc.*, building two piles: face-ups and face-downs, until all the cards in the deck are exhausted and the volunteer has not seen any of his cards.

5. The magician picks up the stack of face-downs and repeats the process, building two piles: *face up, face down, face up, face down, etc.*. (*Note:* The face-ups will be discarded again, so it does not matter whether or not they go onto the pile of face-ups from the previous splitting.)

6. The volunteer still has not seen any of his cards, so the magician picks up the face-downs and repeats the same process again.

7. Finally, the magician picks up the face-downs which are left and repeats the process, leaving only 3 face-down cards, which he turns over to reveal that they are the 3 cards the volunteer had chosen.

[1] `http://www.youtube.com/watch?v=oLjEulT6ssM`

[2] A common variation of this step is to split into stacks of 10, 15, and 15, which requires another step: after putting all the cards back together in step (4), the magician takes the top 4 cards and puts them onto the bottom of the deck. The effect is the same as stacking into 14, 15, and 15 from the start. Although it has a distracting effect on the volunteer, for simplicity we omit this extraneous step.

2 The Trick

The trick to this trick is that the volunteer's random splits have no effect whatsoever on the placement of the 3 cards in the recombined stack. Because of the order of stacking, the stacks which were "split" are put right back together, leaving the first card with 14 cards underneath it, 15 cards between it and the card which went onto the middle stack, and 15 more cards up to the card which went onto the right stack.

The fact that the placement of the cards is always the same, regardless of how the volunteer splits the stacks, is itself the proof that the trick will always work. If it works once (which it does), then repeating it the same way will always provide the desired results, since nothing essential has changed.

This explanation would be sufficient were we to leave it at only proving that this particular trick works. But the question becomes: What mechanism governs the fact that the algorithm works, and is there a way to generalize the trick for any arbitrarily sized deck?

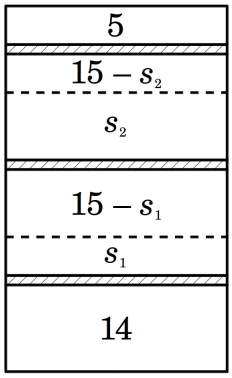

Figure 1: Illustration of stack sizes in between the "final 3" (striped) where s_1 and s_2 are the first and second split sizes, respectively.

3 Trick Mechanics

Notice the cards at odd positions from the top of the deck during the face-up/face-down stacking all get placed face up, while the cards at even positions get placed face down. Then in order to ensure the location of the final three cards to be face down during each repetition of face-up/face-down stacking, they must be at even positions from the top of the deck. But notice that after each application, the position of each card is not only halved but is also reversed so that half of its previous position *from the top* becomes its new position *from the bottom*. For instance, the positions of the 3 cards just before splitting are 6, 22, and 38 (see Figure (1)) from the top. After the first separation, the positions of the 3 cards become 3, 11, and 19, *from the bottom*, out of the new total of 26 cards. To convert the positions from bottom into positions from top, one can simply subtract each position from the size of the deck plus one. Thus the positions 3, 11, and 19 from the bottom out of 26 cards are positions 24, 16, and 8 from the top. Notice these are all even positions, therefore when we split the deck again with the same process, the "final 3" will still remain face-down, this time in positions 12, 8, and 4 from the bottom out of 13 cards, which converts to positions 2, 6, and 10 from the top. Repeating this process, the final 3 go to positions 1, 3, and 5 from the bottom out of 6, or positions 2, 4, and 6 from the top. The last split then leaves them at positions 1, 2, and 3 out of 3, which is of course the desired result.

4 Generalization to Any Size Deck

TERMINOLOGY

m_0	size of starting deck	A	sorting algorithm
m_i	size of deck at step i	$p_{i,j}$	position of j^{th} "chosen" card in deck m_i
n	number of cards volunteer chooses	k	number of times A is applied before arriving at $m_k = n$
j	index of chosen cards, $1 \leq j \leq n$	i	number of times A has been applied
$a_1, a_2, \ldots, a_n$	stack sizes from left to right		

Definition 1. *The* FLOOR FUNCTION *of some real number z, denoted as $\lfloor z \rfloor$, is defined as the greatest integer less than or equal to z.*

Theorem 4.1. *For all nonnegative real numbers z,*

$$\left\lfloor \frac{\lfloor z \rfloor}{2} \right\rfloor = \left\lfloor \frac{z}{2} \right\rfloor. \quad (1)$$

Proof. We can write $z = y + l$ for some nonnegative integer y and some real number $0 \leq l < 1$. Thus,

$$\lfloor z \rfloor = \lfloor y + l \rfloor = y. \quad (2)$$

Substituting (2) into (1) we get

$$\left\lfloor \frac{\lfloor z \rfloor}{2} \right\rfloor = \left\lfloor \frac{y}{2} \right\rfloor. \tag{3}$$

If y is even, then $y/2$ is a nonnegative integer and therefore adding $l/2 \in [0, 1/2)$ will have no effect on the final value after the floor function. Thus, (3) becomes

$$\left\lfloor \frac{\lfloor z \rfloor}{2} \right\rfloor = \left\lfloor \frac{y}{2} \right\rfloor = \left\lfloor \frac{y+l}{2} \right\rfloor = \left\lfloor \frac{z}{2} \right\rfloor.$$

On the other hand, if y is odd, then $y = 2g + 1$ for some nonnegative integer g. Then (3) becomes

$$\left\lfloor \frac{\lfloor z \rfloor}{2} \right\rfloor = \left\lfloor \frac{y}{2} \right\rfloor = \left\lfloor \frac{2g}{2} + \frac{1}{2} \right\rfloor = \left\lfloor g + \frac{1}{2} \right\rfloor = g = \frac{y-1}{2} = \left\lfloor \frac{y-1}{2} \right\rfloor. \tag{4}$$

Since $0 \leq l < 1$, then $\frac{1}{2} \leq \frac{l+1}{2} < 1$ and adding it to the integer $(y-1)/2$ will have no effect after applying the floor function. Thus (4) continues

$$\left\lfloor \frac{\lfloor z \rfloor}{2} \right\rfloor = \left\lfloor \frac{y-1}{2} \right\rfloor = \left\lfloor \frac{y-1}{2} + \frac{l+1}{2} \right\rfloor = \left\lfloor \frac{y+l}{2} \right\rfloor = \left\lfloor \frac{z}{2} \right\rfloor.$$

□

Let A be the sorting algorithm which starts from the top of a deck of cards, turning the top card face up then the next face down into a separate pile, repeating this sequence creating two stacks (one face-up and one face-down) until all of the deck is exhausted. Let m_i be the number of cards in the face-down stack after i applications of A to each face-down stack from $i = 0$ to k. Since any oddly positioned card is discarded into the face-up pile, then if m_i is odd, the remainder after dividing by 2 simply gets discarded. That is, $m_i = \left\lfloor \frac{m_{i-1}}{2} \right\rfloor$. Then with i applications of Theorem 4.1, $m_i = \left\lfloor \frac{m_0}{2^i} \right\rfloor$. Thus k is the number of times the sorting algorithm A is applied to the face-down stacks starting from m_0 cards to leave us with $m_k = n$, which is the number of cards the volunteer would choose from the deck of m_0, and is thus defined as

$$n = \left\lfloor \frac{m_0}{2^k} \right\rfloor.$$

Now let r be the remainder after such a division. That is,

$$m_0 = 2^k n + r, \tag{5}$$

where $0 \leq r < 2^k$. Then after i applications of A to the face-down stacks m_i, we have

$$m_i = \left\lfloor \frac{m_0}{2^i} \right\rfloor = 2^{k-i} n + \left\lfloor \frac{r}{2^i} \right\rfloor. \tag{6}$$

Let $p_{i,j}$ represent the position of the j^{th} "chosen" card after i applications of A for all $1 \leq j \leq n$ and $0 \leq i \leq k$. According to the trick, the desire is that $p_{k,j} = j$. That is, the "final n" cards after k applications of A on the face-down deck are $1, 2, \ldots, n$ from the top of a stack of $m_k = n$ face-down cards.

When A is applied to a face-down stack of m_{i-1} cards with chosen cards at positions $p_{i-1,j}$ then it both halves the positions and also reverses their order out of the resulting m_i face-down cards. Therefore, we can define $p_{i,j}$ recursively as

$$p_{i,j} = m_i + 1 - \frac{p_{i-1,j}}{2}. \tag{7}$$

Note that since we desire $p_{i,j}$ to be an even number at every step (ensuring face-down positioning during A) except at $i = k$, then $p_{i-1,j}/2$ is sure to be an integer for all $0 \leq i \leq k$. Substituting (6) into (7), solving for $p_{i-1,j}$, and simplifying gives us

$$p_{i-1,j} = 2^{k-i+1} n + 2\left\lfloor \frac{r}{2^i} \right\rfloor + 2 - 2p_{i,j}. \tag{8}$$

Ultimately we are interested in $p_{0,j}$, the position of each j^{th} chosen card at the very beginning before applying A. If we know $p_{0,j}$ then we know the sizes $a_1, a_2, \ldots, a_n$ to make the stacks which will allow us to successfully perform the trick.

As we know the final $p_{k,j}$ must be j, then we work a backwards recursion. Thus

$$\begin{aligned} p_{k-1,j} &= 2n + 2\left\lfloor \frac{r}{2^k} \right\rfloor + 2 - 2p_{k,j} \\ &= 2n + 2\left\lfloor \frac{r}{2^k} \right\rfloor + 2 - 2j. \end{aligned}$$

Continuing, we have

$$\begin{aligned} p_{k-2,j} &= 2^2 n + 2\left\lfloor \frac{r}{2^{k-1}} \right\rfloor + 2 - 2p_{k-1,j} \\ &= 2^2 j + 2\left\lfloor \frac{r}{2^{k-1}} \right\rfloor + 2 - 2^2\left\lfloor \frac{r}{2^k} \right\rfloor - 2^2. \\ p_{k-3,j} &= 2^3 n + 2\left\lfloor \frac{r}{2^{k-2}} \right\rfloor + 2 - 2p_{k-2,j} \\ &= 2^3 n - 2^3 j + 2\left\lfloor \frac{r}{2^{k-2}} \right\rfloor + 2 - 2^2\left\lfloor \frac{r}{2^{k-1}} \right\rfloor - 2^2 + 2^3\left\lfloor \frac{r}{2^k} \right\rfloor + 2^3. \\ &\vdots \\ p_{k-i,j} &= 2^i n\left(\frac{1+(-1)^{i+1}}{2}\right) + (-1)^i 2^i j + \sum_{s=1}^{i}(-1)^{s+1}2^s\left(\left\lfloor \frac{r}{2^{k-i+s}} \right\rfloor + 1\right) \\ &\vdots \\ p_{0,j} &= (1+(-1)^{k+1})^k n + (-1)^k 2^k j + \sum_{s=1}^{k}(-1)^{s+1}2^s\left(\left\lfloor \frac{r}{2^s} \right\rfloor + 1\right). \end{aligned} \tag{9}$$

Notice the distance $p_{0,x}$ and $p_{0,x+1}$ between any two consecutive chosen cards x and $x+1$ for $1 \le x < n$ is

$$|p_{0,x+1} - p_{0,x}| = \left|(-1)^k 2^k (x+1) - (-1)^k 2^k x\right| = \left|(-1)^k 2^k\right| = 2^k.$$

Thus, the stack between any two of the final cards is by necessity of the size $2^k - 1$. That is,

$$a_2, a_3, \ldots, a_n = 2^k - 1. \tag{10}$$

The concern then becomes what size a_1 needs to be.

Notice that a_1 must be equal to the size of the original deck m_0 minus the position of the card closest to the *bottom* of the deck. If k is odd, then the card which started at the bottom of the deck would end up at the top of the deck in position $j = 1$ after k applications of A. So if k is odd, we have

$$a_1 = m_0 - p_{0,1}. \tag{11}$$

Substituting (5) and (9) into (11), we get (for odd k),

$$a_1 = r + 2^k + \sum_{s=1}^{k}(-1)^s 2^s\left(\left\lfloor \frac{r}{2^s} \right\rfloor + 1\right). \tag{12}$$

Alternatively, if k is even, then the card which was closest to the bottom at the beginning would end also closest to bottom ($j = n$) after k applications of A. Thus, if k is even, we have

$$a_1 = m_0 - p_{0,n}, \text{ or}$$

$$a_1 = r + \sum_{s=1}^{k}(-1)^s 2^s\left(\left\lfloor \frac{r}{2^s} \right\rfloor + 1\right). \tag{13}$$

Equations (12) and (13) can be combined into one formula for all k as

$$a_1 = r + \left(1+(-1)^{k+1}\right)^k + \sum_{s=1}^{k}(-1)^s 2^s\left(\left\lfloor \frac{r}{2^s} \right\rfloor + 1\right). \tag{14}$$

Separating the k^{th} term from the sum and simplifying, (14) becomes

$$a_1 = r + \left(1+(-1)^k\right)^k + \sum_{s=1}^{k-1}(-1)^s 2^s\left(\left\lfloor \frac{r}{2^s} \right\rfloor + 1\right). \tag{15}$$

5 Final n Trick

We now have all the information we need to perform the trick outlined in section (1) with any size deck.

A magician starts with a deck of m_0 cards. All possible values for the number n of cards for the volunteer to choose are given by

$$\left\{n\colon n = \left\lfloor \frac{m_0}{2^k} \right\rfloor \text{ and } n, k > 0\right\}.$$

The magician should choose one such n and k, then set

$$r = m_0 - 2^k n.$$

The volunteer is asked to remove and memorize n cards from the deck of m_0 without revealing them to the magician. From the remaining cards, the magician makes n stacks of size (from left to right) $a_1, a_2, \ldots, a_n$ given by

$$\begin{aligned} a_1 &= r + \left(1 + (-1)^k\right)^k + \sum_{s=1}^{k-1} (-1)^s 2^s \left(\left\lfloor \frac{r}{2^s} \right\rfloor + 1\right), \text{ and} \\ a_2, a_3, \ldots, a_n &= 2^k - 1. \end{aligned}$$

The volunteer is then asked to place one of his cards on the leftmost stack. Then he is asked to split the second stack anywhere, placing it onto the first, then to place his second card on the second stack. He will be asked to repeat this process until placing his n^{th} card on the n^{th} stack from the left (also the rightmost stack).

The magician then stacks the cards up in the following order. What is left in his hand he stacks onto the rightmost stack, then all of that onto the stack to the immediate left, continuing until all are recombined.

The volunteer is asked to please inform the magician if he sees his card, and so the magician begins the process of splitting the deck face-up then face-down into two stacks, until the deck is exhausted. The volunteer still has not seen his card, so the magician picks up the face-down stack and continues. This process goes on until finally arriving at the n cards which the volunteer had chosen.

6 A Variation of the Sorting Algorithm

The author's variation of the sorting algorithm, which has a different effect on a mathematically-minded audience, is as follows. Instead of face-up/face-down/face-up/face-down/etc., the magician builds three face-down stacks (for illustration let us number the stacks 1,2,3) in the order 1,2,3,2,1,2,3,2,1,2,3,2,.... Notice that stack 2 will contain the exact same cards as the face-down stack of the original sorting algorithm and will also have about twice as many cards as each of the other two stacks. Also note that the other two stacks contain the same cards as the discarded face-ups from the original sorting algorithm.

From the beginning of the trick to the end, the magician rattles on about how this trick is all about probability, and each time he builds the stacks he asks the volunteer in which stack it is most likely to find at least one of his cards. Of course, most people will understand the bigger stack in the middle is the most likely to find at least one of their cards. So the magician discards the other two stacks and does the same sorting algorithm with the middle stack. Again, the magician lets the volunteer choose the "most likely" stack to contain at least one of his cards. Choosing the bigger stack, again the other two stacks are discarded onto the cumulative discard pile. Repeating this until there is only one sorting algorithm remaining before arriving at the volunteer's chosen cards, the magician makes a scene about how the stack is getting very small and he wants to really get an idea if one of the volunteer's cards are in there. So the magician reverts to the original algorithm of face-up, face-down, face-up, ..., asking the volunteer to please let him know if he sees one of his cards. Not seeing any of his cards, the volunteer may think the trick has failed.

Now there is a huge discard pile and a small stack the size of however many cards the volunteer had chosen. Now the same question can be asked of the volunteer: "Which stack is more likely to contain at least one of the chosen cards?" Depending on the volunteer and the magician, the banter can go many ways. Of course then all the chosen cards are revealed to be in the small stack.

Note that in the author's experience, this variation, while amusing to mathematicians who can quickly see that it has absolutely nothing to do with probability, is not amusing to non-mathematically minded individuals who merely respond, "Oh, I was never good at probability in school." For the general population, the original trick has proven to the author to be much more impressive, but with mathematicians the variation can be quite amusing.

7 Generating Results

A few lines of code can easily be implemented to produce results for various deck sizes. The following code was run in one line in the command line of PARI/GP to produce a txt file with all the possible ways to perform the card trick with deck size ranging from 7 to 500.

```
write("card_magic.txt", "m=deck size\nn=number of cards to choose\n
a1--an=stack sizes from left to right\n------------------------");
for(m=7,500,write("card_magic.txt","m=",m);
for(k=2,20,if(floor(m/2^k)==0, break,
n=floor(m/2^k);
r=m-2^k*n;a=r+(1+(-1)^k)^k+sum(j=1,k-1,(-1)^j*2^j*(floor(r/2^j)+1));
if(n>1,write("card_magic.txt","  n=",n,", a1=",a,", a2--an=",2^k-1),
write("card_magic.txt","  n=",n,", a1=",a)))))
```

The output file card_magic.txt can be downloaded at http://ajbertagnolli.com/card_magic.txt.

8 Conclusion

Now that the "Final 3" trick has been explained mathematically and generalized for any size deck, variations on the trick are limited only by the magician's imagination. For instance, using the computer-generated results from section (6), we can build one variation of the trick by letting the volunteer choose a number between 2 and 6 and then have them remove that many cards from the deck and memorize them. Then, looking at our results we can find the largest deck less than or equal to 52 which will allow for the trick to be performed with the number of cards chosen by the volunteer. Then, keeping our deck of 52, we can simply do the trick as though working with the smaller deck and its corresponding stack sizes, which simply means we will have some extra cards at the top after stacking them back together (step (4) of section (1)). Then before beginning the first application of A, we simply discard (in whatever fashion we would like) as many "extra" cards as we had on top.

For example, if the volunteer chooses 5 cards, we would look at our generated results for the largest number less than or equal to 52 in which the trick can be performed with $n = 5$ cards. The appropriate section of the results reads:

```
m=47
  n=11, a1=3, a2--an=3
  n=5, a1=7, a2--an=7
  n=2, a1=14, a2--an=15
  n=1, a1=14
```

where m is the deck size, n is the number of cards chosen, a1 is the leftmost stack size, and a2--an are the remaining n-1 stack sizes.

So we make 5 stacks of size 7 and do the trick as explained elsewhere in this paper, with one exception: since m is 47 but we are using a deck of 52, we must discard the top 5 cards before the first application of A, then we just continue the trick as normal. With the added level of randomness in being able to choose any number of cards instead of only 3, the volunteer will be even more impressed when we reveal that the final 5 cards are the very ones which he had chosen.

Acknowledgments I extend my thanks to Dr. Luis Cueva-Parra for pointing out to me that $(k + 1) \bmod 2 = (1 + (-1)^k)/2$. This is a very useful relation that simplified calculations in this paper and eliminated the need for modular notation. I also would like to thank Dr. Enoch Lee for his indispensable advice on writing proper mathematics.