Magic: discover 6 cards from a 52-card deck knowing the 6 colors based on the De Bruijn sequence

Siang Wun Song
Institute of Mathematics and Statistics
University of SÃ£o Paulo

In Memoriam
Marco Dimas Gubitoso

Abstract

We present a magic in which a deck of 52 cards is cut any number of times and six cards are dealt. By only knowing their colors, all six cards are revealed. It is based on the De Bruijn sequence. The version of telling 5 cards from an incomplete 32-card deck is well known. We adapted it to the 52-card version, by obtaining a 52-bit cyclic sequence, with certain properties, from a 64-bit De Bruijn sequence. To our knowledge, this result is not known in the literature. The rest of the paper presents the De Bruijn sequence and why the magic works.

1 Introduction

We present the following magic. The magician takes a complete deck of 52 cards. After cutting the deck of cards several times, 6 participants are chosen and a card is dealt to each one. The magicion asks who holds a red card. With the aid of a slide presentation file, the magicion inputs the 6 colors into the slide file that reveals the 6 cards dealt. The reader can go directly to section 2 to practice the magic trick and then go back to the rest of the paper to understand why it works.

The magic is based on the De Bruijn sequence [4, 12]. The magic of discovering 5 cards in an incomplete deck of 32 cards, which we will call **Magic Version 32**, is well known [2, 7]. It uses a 5-bit string to encode each card: the first bit represents the color, the first 2 bits encode the suit and the last 3 bits the value of the card. A De Bruijn sequence of order 5 is used to generate the 32 5-bit strings.

A magic that uses a full deck of 52 cards to reveal 6 cards can be found on The Aperiodical and Github websites [13, 14]. However, the magic does not allow *wrap-around*, that is, it is not possible to cut the deck of cards any number of times before dealing the six cards. The reason is that the magic is based on a non-cyclic 57-bit sequence, whose 52 6-bit subsequences encode the 52 cards.

We will denote by **Magic Version 52** the magic that uses the complete deck of 52 cards, which can be cut as many times as needed, before dealing the 6 cards. We need now 6 bits to encode each card. The coding of color and suit is as before, but we need 4 bits to encode the card value. Magic Version 52 is based on a 52-bit cyclic sequence, extracted from a 64-bit De Bruijn sequence. We will show that some of the coded card values do not correspond to playing card values. These constitute in card value coding exceptions. We found several attempts to use this strategy.

One such 52-bit sequence is presented on The Magic Café website [15, 14], with authorship attributed to Larry Finley. However, the 52-bit sequence has 28 ones and 24 zeros, which would require the use of two wild cards or Jokers in place of two normal cards of the same color.

Another 52-bit sequence, mentioned in a post on Github [14], has 26 ones and 26 zeros, thus avoiding the use of two Jokers. This sequence, however, does not have the same number (i.e. 13) of subsequences of length 2. It has 12 00s, 14 01s, 14 10s and 12 11s. Therefore, it needs to allow exceptions to encode some card suits.

In the Magic Version 52 presented here, we were able to extract a 52-bit cyclic string from a 64-bit De Bruijn sequence, which has an equal number (13) of 2-bit subsequences (00, 01, 10 and 11). This is a strong condition, since it implies the weaker condition of equal number (26) of ones and zeros. With that, we avoid having exceptions to encode the card suits. To our knowledge, obtaining a sequence with these characteristics is original, not known in the literature. Furthermore, with the use a slide presentation file that stores card value coding exceptions (see later sections), there is no need to memorize any information. With that, even a child can present this magic.

Section 2 presents the preparation and a presentation of the magic. Once motivated, the reader will be curious to know why the magic works. In Section 3 we introduce the concept of the De Bruijn sequence that is used in the magic. Section 4 explains Magic Version 32: discover 5 cards from an incomplete deck of 32 cards. Section 5 adapts the same idea of the previous section to Magic Version 52: discover 6 cards in a complete 52-card deck. Finally in Section 6 we comment on the presented magic, and present some applications of De Bruijn sequences and De Bruijn graphs in different areas of Science.

2 Preparation and presentation of the magic

Let us simulate a presentation of the magic. Once impressed and motivated, you will be curious to want to know why this works. We will then present the detailed explanation in the following sections.

2.1 Preparation

You will need a complete deck of 52 cards and a slides presentation file (magic52.pdf) that is used to present the magic.

Download the magic52.pdf slide file:

https://www.ime.usp.br/∼song/magic/magic52.pdf

Prepare the 52-card deck by placing the cards in a determined order. To obtain this order, use Figure 8 (page 8). The first card in the figure is 2♥. Place this card as the first card. The second card in Figure 8 is Ace♥. Put it under the first one. Likewise arrange the remaining cards. The last card must be Ace♠. The prepared deck is shown in Figure 1.

2.2 Presentation of the magic

Take the prepared deck and, before the audience, cut the cards as many times as you wish.

Cutting the deck of cards means separating the deck into two parts and put the top part under the bottom part. See Figure 2.

Choose 6 participants and deal a card to each one.

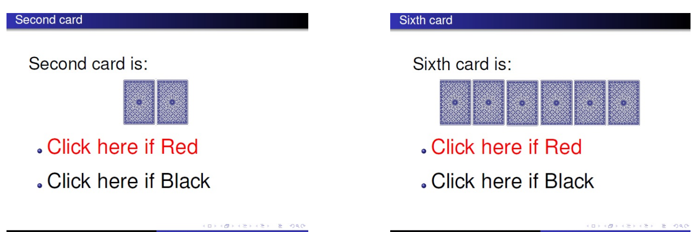
Ask the 6 participants who has a red card, that is, with ♥ (Hearts) or ♦ (Diamonds) suit.

Open the slide file (slide number 3), enter the card color of the first participant.

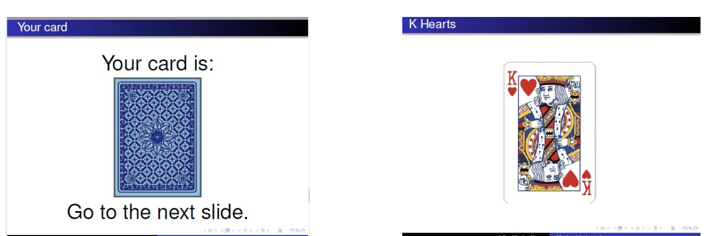
Figura 1: The prepared deck of cards.

Figura 2: Cutting a deck of cards.

Then input the card colors, one by one, of the remaining five participants, into the slide file.



Create some suspense and, to everyone's amazement, move to the next slide that will show the first participant's card.



Move to the next slides to reveal the other participants' cards.

In the remaining sections, we explain why this magic works.

3 De Bruijn sequence

A De Bruijn binary sequence of order n, denoted by $B(n)$, is a cyclic sequence of 2^n zeros and ones where each possible n-bit string occurs exactly once as a substring of the sequence $B(n)$.

For example, for $n = 5$, a De Bruijn $B(5)$ sequence is shown in the following.

$$\underbrace{00001}_{\text{string1}}001011001111100011011101010$$

$$0\underbrace{00010}_{\text{string2}}01011001111100011011101010$$

From this sequence, one can obtain all the 32 5-bit strings, from 00000 to 11111. See Figure 3. The 5-bit string in row 1 (containing 00001) corresponds to the first 5 bits of the De Bruijn $B(5)$ sequence. The 5-bit string in row 2 (containing 00010) corresponds to the second bit up to the sixth bit of the sequence, and so on. Recall that the sequence is cyclic, i.e. after the last bit follows the first. Thus, for example, the string 00000 is obtained with the last bit 0 followed by the first 4 bits 0 of the sequence.

1.	00001	17.	11000
2.	00010	18.	10001
3.	00100	19.	00011
4.	01001	20.	00110
5.	10010	21.	01101
6.	00101	22.	11011
7.	01011	23.	10111
8.	10110	24.	01110
9.	01100	25.	11101
10.	11001	26.	11010
11.	10011	27.	10101
12.	00111	28.	01010
13.	01111	29.	10100
14.	11111	30.	01000
15.	11110	31.	10000
16.	11100	32.	00000

Figura 3: All the 32 5-bit strings.

By the way these 5-bit strings are obtained from the De Bruijn sequence, we can notice several properties. One is that the sequence itself is reproduced in the first column of Figure 3. Another important property is the following. On each row, for example on row 10 of Figure 3, the 5-bit string (11001) also appears in the first column of the 5 rows, from row 10 through row 14 (11001).

As the De Bruijn sequence is cyclic, the same observation applies also, for example, on row 31 with the string 10000. This property will be used in the magic, as we will see in the next section.

4 Magic Version 32: discover 5 from 32 cards

As previously stated, the magic of revealing 5 cards from an incomplete deck of 32 cards, knowing their colors, is well known [2, 7]. We now explain how it works.

A 5-bit string is used to encode each card. The first bit encodes the color of the card, as follows:

0	Red
1	Black

The first two bits encode the suit of the card. See Figure 4.

00	Hearts	♥
01	Diamonds	♦
10	Spades	♠
11	Clubs	♣

Figura 4: First two bits encode the suit of the card.

000	8
001	A
010	2
011	3
100	4
101	5
110	6
111	7

Figura 5: Last three bits encode the card value.

The last 3 bits of the string encode the value of the card (see Figure 5). Only the Ace card up to 8 are used. (Some magicians prefer to use the Ace and the 7 through K cards. To do this, simply add 7 to the value corresponding to the 3 bits to facilitate memorization, adopting the value 14 to represent Ace.)

Using this encoding and Figure 3, the 32 cards are encoded in Figure 6.

1.	00001	A♥	17.	11000	8♣
2.	00010	2♥	18.	10001	A♠
3.	00100	4♥	19.	00011	3♥
4.	01001	A♦	20.	00110	6♥
5.	10010	2♠	21.	01101	5♦
6.	00101	5♥	22.	11011	3♣
7.	01011	3♦	23.	10111	7♠
8.	10110	6♠	24.	01110	6♦
9.	01100	4♦	25.	11101	5♣
10.	**11**00**1**	A♣	26.	11010	2♣
11.	**1**0011	3♠	27.	10101	5♠
12.	00111	7♥	28.	01010	2♦
13.	01111	7♦	29.	10100	4♠
14.	**1**1111	7♣	30.	01000	8♦
15.	11110	6♣	31.	10000	8♠
16.	11100	4♣	32.	00000	8♥

Figura 6: The 32 encoded cards.

Consider again row 10 of Figure 6. Using the property noted in the previous section, the 5 bits of row 10 are equal to the first bits of the five rows starting from row 10. Since the first bit of each card represents its color, we can obtain the 5-bit code for each row, which represents a card (color, suit and value), by only knowing the colors of five consecutive cards. Another fact is that the 5-bit codes for all rows are distinct, by construction. That is the explanation of why the magic works.

Once knowing the first card, the following 4 cards can be obtained, by memorizing the sequence. However, the 5-bit strings in Figure 6 have a useful property: knowing the value of a 5-bit string, one can easily obtain the value of the next string, in the following way. Let the 5 bits of a string be $b_1b_2b_3b_4b_5$. The next string has the first 4 bits equal to $b_2b_3b_4b_5$ and the fifth bit is equal to $b_1 \oplus b_3$ where $\oplus$ is the exclusive-or operation. This property, however, does not apply if the string is 00000, as the next string would also be 00000 by the rule. The solution that magicians found is simply by desconsidering the 00000 string (in our coding by removing the 8♥ card).

5 Magic Version 52: discover 6 from 52 cards

In the previous section, we used a 32-bit De Bruijn sequence $B(5)$ to generate the 32 distinct 5-bit strings. Each 5-bit string is used to encode one of the 32 cards.

In this section, we show how to adapt the same ideas to the magic of discovering 6 cards in a complete deck of 52 cards.

5.1 Preliminary ideas

For 52 cards, we will encode each card with a 6-bit string. The coding of color (red or black) and of the card suit follows the same rules as in the previous section. The last 4 bits encode the card value (from Ace to K), as follows. Note that codes 0000, 1110 and 1111 (marked with *) do not correspond to the value of any card. There are therefore 12 6-bit strings that do not correspond to cards from the deck.

0000	*
0001	A
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	J
1100	Q
1101	K
1110	*
1111	*

For Magic Version 52, we will eliminate 12 bits from a 64-bit De Bruijn $B(6)$ sequence, with the following conditions.

1. In the new 52-bit (cyclic) sequence, each 6-bit subsequence must appear only once.
2. The new 52-bit sequence has an equal number of subsequences 00, 01, 10 and 11.

Condition 1 is necessary because each 6-bit subsequence must determine a unique card.

Note that the card suit is encoded by 2 bits, so in condition 2 we want to have the same number of each suit, that is, 13 of each suit.

We show below that it is possible to obtain such a 52-bit sequence with the two conditions above.

5.2 Obtaining a new 52-bit sequence

We consider a De Bruijn sequence $B(6)$, called CCR1, from the paper by Daniel Gabric, Joe Sawada, Aaron Williams and Dennis Wong [4].

CCR1: 0000001111110001101110011001010110101001000100111011000010111101

The new sequence that satisfies the two conditions is obtained by removing the 12-bit portion, as shown in the next figure.

$$000000111111000110111001100101011010100100010011101\underbrace{100001011110}_{Remove}1$$

We thus obtain the following new sequence.

$$0000001111110001101110011001010110101001000100111011$$

From this 52-bit (cyclic) sequence, we obtain 52 6-bit strings, along with the encoded cards. See Figure 7. The strings marked with an asterisk (*) do not code valid cards.

1	000000*		27	010101	5♦
2	000001	A♥	28	101011	J♠
3	000011	3♥	29	010110	6♦
4	000111	7♥	30	101101	K♠
5	001111*		31	011010	10♦
6	011111*		32	110101	5♣
7	111111*		33	101010	10♠
8	111110*		34	010100	4♦
9	111100	Q♣	35	101001	9♠
10	111000	8♣	36	010010	2♦
11	110001	A♣	37	100100	4♠
12	100011	3♠	38	001000	8♥
13	000110	6♥	39	010001	A♦
14	001101	K♥	40	100010	2♠
15	011011	J♦	41	000100	4♥
16	110111	7♣	42	001001	9♥
17	101110*		43	010011	3♦
18	011100	Q♦	44	100111	7♠
19	111001	9♣	45	001110*	
20	110011	3♣	46	011101	K♦
21	100110	6♠	47	111011	J♣
22	001100	Q♥	48	110110	6♣
23	011001	9♦	49	101100	Q♠
24	110010	2♣	50	011000	8♦
25	100101	5♠	51	110000*	
26	001010	10♥	52	100000*	

Figura 7: The 43 encoded cards. (*) indicates coding of invalid card.

Notice that in Figure 7, the following 9 cards are missing: 4♣, 10♣, K♣, 7♦, 2♥, 5♥, J♥, 8♠ and A♠. On the other hand, we have 9 strings marked with an asterisk (*) (with the last four bits equal to 0000, 1110 or 1111) that do not correspond to playing card values. Counting the strings marked with (*), we have exactly three with the first two bits encoding ♣, one encoding ♦, three encoding ♥, and two encoding ♠.

Thus, we use the cards marked with (*) to encode the missing cards, as exceptions. This is shown in Figure 8. The magician would have to memorize these exceptions. However, as we saw in Section 2, there is no need to memorize anything by using a presentation file that stores all the codings and exceptions.

6 Final comments

The card magic presented here uses the full 52-card deck. It allows you to cut the cards as many times as necessary before dealing the 6 cards. To make this magic possible, the difficulty lies in extracting a 52-bit cyclic string, from a 64-bit De Bruijn sequence of order 6, which satisfies the two conditions of Section 5.1. To our knowledge, we found no other works with these characteristics in the literature.

1	000000*	2♥	27	010101	5♦
2	000001	A♥	28	101011	J♠
3	000011	3♥	29	010110	6♦
4	000111	7♥	30	101101	K♠
5	001111*	5♥	31	011010	10♦
6	011111*	7♦	32	110101	5♣
7	111111*	4♣	33	101010	10♠
8	111110*	10♣	34	010100	4♦
9	111100	Q♣	35	101001	9♠
10	111000	8♣	36	010010	2♦
11	110001	A♣	37	100100	4♠
12	100011	3♠	38	001000	8♥
13	000110	6♥	39	010001	A♦
14	001101	K♥	40	100010	2♠
15	011011	J♦	41	000100	4♥
16	110111	7♣	42	001001	9♥
17	101110*	8♠	43	010011	3♦
18	011100	Q♦	44	100111	7♠
19	111001	9♣	45	001110*	J♥
20	110011	3♣	46	011101	K♦
21	100110	6♠	47	111011	J♣
22	001100	Q♥	48	110110	6♣
23	011001	9♦	49	101100	Q♠
24	110010	2♣	50	011000	8♦
25	100101	5♠	51	110000*	K♣
26	001010	10♥	52	100000*	A♠

Figura 8: The 52 encoded cards. (*) indicates an exception in the card value coding.

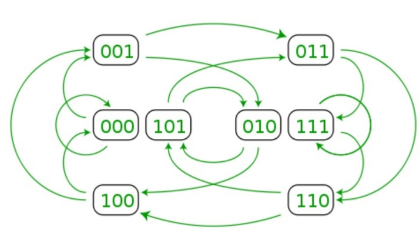
Let us now comment on the two conditions of Section 5.1.

1. In the new 52-bit (cyclic) sequence, each 6-bit subsequence must appear only once.
2. The new 52-bit sequence has an equal number of subsequences 00, 01, 10 and 11.

Recall that we use a 6-bit string to encode each card: the first bit to encode the color, the first two bits the card suit, and the four last bits the card value. Condition 1 is necessary because each 6-bit subsequence must determine a unique card. Another necessary condition is that the 52-bit sequence must present equal number (i.e. 26) of bits 0 and 1, since we have 26 red cards and 26 black cards. On the other hand, we have 13 cards with the same suit. Condition 2 ensures that we can encode the card suits adequately. Condition 2 is a strong condition since it garantees the same amount (26) of 0s and 1s. Note that the 4-bit subsequences of the 52-bit sequence have values from 0000 to 1111. Therefore we have to allow exceptions in the coding of card values.

The discovery of the De Bruijn sequence was due to Nicolaas Govert de Bruijn (1918 - 2012), a Dutch mathematician of the Eindhoven University of Technology. We briefly present below some applications of the De Bruijn sequence.

As before, we consider binary sequences. The De Bruijn sequence is related to the De Bruijn graph. A De Bruijn graph of dimension n is a directed graph where the vertices are all 2^n n-bit strings. Each vertex has two directed edges, defined as follows. A vertex $b_1 b_2 \ldots b_n$ has directed edges to the vertices $b_2 \ldots b_n 0$ and $b_2 \ldots b_n 1$. Example: the folllowing figure shows a De Bruijn graph of dimension 3.



Source: Wikipedia

A De Bruijn sequence of order n can be constructed by a Hamiltonian path of a De Bruijn graph of dimension n. Consider for example the De Bruijn graph of dimension 3 in the figure. A Hamiltonian path is as follows.

000
001
010
101
011
111
110
100

A De Bruijn sequence of order 3 is obtained by considering the first bit of each string of the figure above: 00010111. There is another possible De Bruijn sequence of order 3, namely, 11101000.

One use of the De Bruijn graph is in Parallel Computing when we want to find interconnection networks that minimize the distance between two computers. Diameter of a graph is the longest distance between any two vertices. For example, by interconnecting N computers in a ring network, we will have a diameter of $N/2$. The diameter of a De Bruijn graph of dimension n, with 2^n vertices, is n. Thus, by interconnecting N processors in the form of a De Bruijn graph, the diameter is $\log N$. C. Lavault [6] discusses this issue considering a De Bruijn graph with a non-binary alphabet (each vertex has directed edges to $k > 2$ other vertices). For N vertices or computers, the diameter is $\log_k N$.

In Bioinformatics, the De Bruijn graph is used to the assembly of DNA fragments and for *de novo* assembly of nucleotide sequences [9, 1]. The De Bruijn graph is also used in the Koorde protocol for distributed hash tables [5].

The book by Persi Diaconis and Ron Graham [2] presents several applications, in Robotic Vision, making codes to encrypt data, and DNA analysis.

We refer the interested reader to a large amount of material on De Bruijn sequence and graph in the literature e.g. [3, 8, 10, 11].

Referências

[1] R. Chikhi, A. Limasset, S. Jackman, J. Simpson, P. Medvedev. On the representation of de Bruijn graphs. RECOMB 2014: Proceedings of the 18th Annual International Conference on Research in Computational Biology - Volume 8394, April 2014, pages 35-55.

[2] Persi Diaconis and Ron Graham. *Magical Mathematics: The Mathematical Ideas That Animate Great Magic Tricks*. Princeton University Press, 2012.

[3] Hal Fredricksen, and Anthony Ralston. De Bruijn Sequences- A Model Example of the Interaction of Discrete Mathematics and Computer Science. Mathematics Magazine 55, no. 3 (1982): 131-143.

[4] Daniel Gabric, Joe Sawada, Aaron Williams and Dennis Wong. A framework for constructing de Bruijn sequences via simple successor rules. *Discrete Mathematics* 341(11):2977-2987, November 2018.

[5] M. Frans Kaashock, David R. Karger. Koorde: a simple degree-optimal distributed hash table. International Workshop on Peer-to-Peer Systems. IPTPS 2003, pp. 98-107. LNCS Volume 2735, Springer-Verlag.

[6] Christian Lavault. Interconnection Networks: Graph- and Group Theoretic Modelling. HAL - a multi-disciplinary open access archive. Submitted on 8 Mar 2010. https://hal.archives-ouvertes.fr/hal-00461923

[7] F. Thompson Leighton. *Introduction to parallel algorithms and architectures: arrays, trees, hypercubes*. Morgan Kaufmann Pub. 1992. (See page 487.)

[8] Antoine Limasset, Bastien Cazaux, Eric Rivals and Pierre Peterlongo. Read mapping on de Bruijn graphs. *BMC Bioinformatics* 17, Article number 237. 2016.

[9] Pavel A. Pevzner, Haixu Tang, and Michael S. Waterman. An Eulerian path approach to DNA fragment assembly. PNAS August 14, 2001 98 (17) 9748-9753.

[10] Joe Sawada, Aaron Williams, Dennis Wong. A surprisingly simple de Bruijn sequence construction. *Discrete Mathematics* 339(1):127-131, Januray 2016.

[11] Chi Him Wong. Novel universal cycle constructions for a variety of combinatorial objects. Ph.D. thesis in Computer Science. The University of Guelph. 2015. 117 pages.

[12] Shenquan Xie. Notes on de Bruijn sequences. *Discrete Applied Mathematics* 16 (1987) 157-177.

[13] https://aperiodical.com/2012/08/davids-de-bruijn-sequence-card-trick/

[14] https://gist.github.com/christianp/3206992

[15] https://www.themagiccafe.com/forums/viewtopic.php?topic=676384