Contents

Card trick instructions and pseudocode

Summary

The trick described here requires no sleight of hand or misdirection, but depends only on following (strictly) a sequence of steps to arrive at a counterintuitive (but mathematically provable) result.

Video

https://www.youtube.com/watch?v=aNpGxZ_1KXU

Instructions

Starting conditions

- *Deck* of playing cards:
  - Shuffled or unshuffled.
  - Need not be a full deck (i.e. can be a partial deck, or even multiple decks), but must have an equal number of black and red cards.
  - Initially facedown in a single pile.

Steps

Divide cards into piles

1. Until the *deck* is empty, do the following:

    1. Draw the top card, and turn it faceup to reveal its color. We'll refer to this card as the *indicator*.

        - If the *indicator* card is black,

            – draw the *next* card from the *deck*, and place it (still facedown) in the *black pile*;

          otherwise,

            – draw the *next* card from the *deck*, and place it (still facedown) in the *red pile*.

    2. Place the *indicator* card in discard pile; it won't be used in the remaining steps.

Swap cards between piles

1. Count the number of cards in the *black pile* and in the *red pile*. (Note: These counts are not necessarily equal.)
2. Pick (by any desired means) a number between 0 and the smaller of the *black pile* and *red pile* counts, inclusive; call this the *swap count*.
3. Select (by any desired means) *swap count* cards from the *black pile* and the same number from the *red pile*.
4. Add the *swap count* cards taken from the *black pile* to the *red pile*, and the *swap count* cards taken from the *red pile* to the *black pile*.
5. Count the number of black cards in the *black pile*; call this *black count*.
6. Count the number of red cards in the *red pile*; call this *red count*.

Tally the results

1. Compare *black count* and *red count*. If the steps above have been followed correctly, **these counts will always be equal.**

Pseudocode

This expression of the procedure relies heavily on mathematical notation, with key symbols and functions explained in "Definitions". Most of the steps are accompanied by additional explanatory text (in parentheses).

Definitions

- The $\in$ symbol ("is an element of") indicates membership in a set or sequence. For example, $v \in S$ indicates that the object v is in the set or sequence S. (Note that our usage of this notation doesn't indicate whether S is a set or a sequence; instead, that will be inferable from the context.)
- The $\forall$ symbol ("for all") indicates that the mathematical or logical relationship stated before the symbol holds for all values following the symbol.

  For example,

  $$x^2 \leq 100, \forall x \in \{-10, -9, \ldots, 0, \ldots, 9, 10\}$$

  states that the relationship $x^2 \leq 100$ holds for all integer x values between -10 and 10, inclusive.
- The assignment operator used here is $\leftarrow$ ("gets"): The expression on the right-hand side of the operator is evaluated, and the result is assigned to the variable on the left-hand side.
- The $\frown$ operator is used for concatenation of sequences. So $A^\frown B$ is a sequence containing all of the elements of A, followed by the elements of B: $(a_0, a_1, a_2, \ldots, b_0, b_1, b_2, \ldots)$.
- $\mathrm{len}(S)$ is a function that returns the length of the sequence S.
- $\mathrm{head}(S, n)$ is a function that returns a sequence containing the first n elements of the sequence S—that is, $(s_0, s_1, \ldots, s_{n-1})$.
- $\mathrm{tail}(S, n)$ returns a sequence of all elements of S *after* the first n elements—i.e. $(s_n, s_{n+1}, \ldots)$.
- $\mathrm{color}(v)$ is a function that returns the *color* of some object v.
- $\mathrm{filter}(S, c)$ returns a sequence containing all elements $d \in S$ with $color(d) = c$.
- $\mathrm{rand}(n)$, where $n \in \{0, 1, \ldots\}$, returns a randomly selected integer between 0 and n, inclusive. (No particular distribution is specified. For example, a uniform distribution, where each of the values is equally likely to be selected, may be used—but it is neither required nor assumed.)
- $\min(a, b)$ the lesser of the two values a and b. A more formal way of stating this is

  $$\min(a, b) = \begin{cases} a & \text{if } a \leq b; \\ b & \text{otherwise.} \end{cases}$$

Inputs

- D (the *deck*) is a sequence of elements, $D = (d_0, d_1, d_2, \ldots)$, such that:
  - $\text{color}(d) \in \{black, red\}, \forall d \in D$.

    (The color function returns *black* or *red* for every element of D.)
  - $\text{len}(\text{filter}(D, black)) = \text{len}(\text{filter}(D, red))$.

    (There are equal numbers of *red* and *black* elements.)

Declarations

Note: Temporary variables aren't included in this section; instead, they are indicated in bold type on their first usage. (The type of any such variable should be inferable by the reader from the context.)

- $B \leftarrow ()$.

  (B is a sequence, initially empty; this represents the *black pile*.)
- $R \leftarrow ()$.

  (R is a sequence, initially empty, representing the *red pile*.)

Steps

Assign elements to separate sequences

1. While $\text{len}(D) \neq 0$:
   1. $\mathbf{p} \leftarrow d_0$.

      (Element 0 of D is assigned to p, which represents the *indicator* card.)
   2. $\mathbf{q} \leftarrow d_1$.

      (Element 1 of D is assigned to q, the *next* card after the *indicator*.)
   3. $D \leftarrow \text{tail}(D, 2)$.

      (Remove the first two elements, p and q, from D.)
   4. if $\text{color}(p) = black$:
      - $B \leftarrow B^\frown(q)$;

        (Append q to B.)

      otherwise:
      - $R \leftarrow R^\frown(q)$.

        (Append q to R.)

Swap elements between sequences

1. $\mathbf{u} \leftarrow \min\big(\text{len}(B), \text{len}(R)\big)$.

   (Assign the smaller of the lengths of B and R to u.)

2. $\mathbf{n} \leftarrow \text{rand}(u)$.

   (Assign to n an integer randomly selected from $\{0, 1, \ldots, u\}$.)

3. Reorder the elements within the sequences B and R in any way desired (or not at all), without moving any elements from one sequence to the other.

4. $\mathbf{S_B} \leftarrow \text{head}(B, n)$.

   (Assign to S_B the sequence containing the first n elements of B.)

5. $\mathbf{S_R} \leftarrow \text{head}(R, n)$.

   (Assign to sequence S_R the sequence containing the first n elements of R.)

6. $B \leftarrow \text{tail}(B, n)^\frown S_R$.

   (Assign to B the concatenation of the contents of B following the first n elements with S_R.)

7. $R \leftarrow \text{tail}(R, n)^\frown S_B$.

   (Assign to B the concatenation of the contents of R following the first n elements with S_B.)

Filter sequences and compare lengths

1. $\mathbf{c_b} = \text{len}(\text{filter}(B, \mathit{black}))$.

   (Assign to c_b the count of elements $b \in B$ with $\text{color}(b) = \mathit{black}$.)

2. $\mathbf{c_r} = \text{len}(\text{filter}(R, \mathit{red}))$.

   (Assign to c_r the count of elements $r \in R$ with $\text{color}(r) = \mathit{red}$.)

3. Verify that $c_b = c_r$.

   (If the steps above have been followed correctly, **these counts will always be equal.**)