**Algorithms Notes**

# The Card Flip Game

By Jonathan Liu

Updated: 4/14/21

## 1 The Game

The rules of the game are simple. I start by shuffling a standard deck of cards. Then, I take a card off the top of the deck, show it to you, and do the same with the next card, repeating until there are no more cards left in the deck. At any point in this process (even before I show you the first card), you can tell me to stop. At that point, if the next card I flip over is red, you win. If it's black, you lose.

The question is, what's the optimal strategy for this game, and what is its winning probability?

You can assume the game is fair and in good faith, but to avoid spoilers and appease any cheeky readers I'm going to fill the rest of this page with a bunch of (basically meaningless) details. I would strongly recommend taking your time to think about it before moving on! If this were a classroom setting, I'd give you $\tilde{2}$0 minutes or something - finding your strategy's winning probability can take a while. The answer is on the next page, so stop here if you don't want to see it yet.

### 1.1 The Pedantic Details

- The deck consists of 52 cards, with 26 red cards and 26 black cards. Each card is either red or black, and no cards are both red and black.
- Each card appears perfectly identical to the rest of the deck prior to being flipped over.
- After each card is flipped over, you can see its color, and you may use that information to affect your decisions.
- You're maximizing your probability of victory in the long run, so your strategy should optimize your winning probability over all shuffles. With that in mind, you can expect the deck to be fully random (each ordering of the cards is equally likely to occur after each shuffle).
- To be clear, this also means that each shuffle is completely independent of all prior shuffles.
- If there is only one card left in the deck, you are required to "select" that card.
- The dealer will not know the order of the cards, and as such can not be blackmailed, bribed, coerced, or mind-read into revealing the color of the next card.
- I'm pretty confident that the above information is enough to ensure that you can gain no advantage from any other information related to the deck, shuffle, or dealer. However, just to be absolutely explicit, the only information you're allowed to use at any point to make your decision is the colors of the cards that have already been flipped over, if any.

# 2 The Solution

Well, there's good news and there's bad news. Good news first: the strategy you thought of is the best possible one! Your brilliance really shone through here, most likely a direct result of my incredible guidance. I knew you had it in you.

Now, the bad news: the strategy you thought of is literally the worst one imaginable. There doesn't exist a single strategy that does worse than yours in the long run, so it would've saved you so much time and energy if you had just not given any thought to this problem. A goldfish playing this game would have done just as well as you would.

Wait, what?

## 2.1 A Bold Claim

All joking aside, the statements I made earlier were both true, which only leads to one possible conclusion: every algorithm has the exact same probability of success. Which is a pretty wild realization for a game where it seems like so much depends on the information you have ass the game goes on. We know that stopping before the first card gives a success probability of $1/2$, which leads to the following claim:

**Claim 1.** *Every legal strategy to play this game has a success rate of exactly* 50%.

Before we dive into the proof, we should start with an example. Some algorithm that isn't trivial, but isn't too complicated either.

**Example 2.** Let's call this strategy "Safe Bet": you only stop when you know for sure the next card will be red, which means you've seen all the black cards be flipped already. Or, if there's one card left, you have to stop there, regardless of what the last card's color it.

**Algorithm 1** SAFE BET

```
procedure STRATEGY
    if There are only red cards left in the deck then
        Say "stop."
    else if There is only one card left then
        Say "stop."
    else
        Continue.
    end if
end procedure
```

Now, we'll analyze this procedure two different ways. The first way is perhaps the more straightforward way: finding the probability that all the black cards run out before the red cards do. Note that either the red cards or the black cards must run out first, and that one of the two must occur, so these events are both mutually exclusive and have probability summing to 1. But because there are the same number of black and red cards, these events are symmetrical, so the probability of black cards running out before red cards do is exactly $1/2$. (You can also argue through one-to-one correspondence, or through many other more tedious methods.)

The second way we can analyze this procedure is by reframing the algorithm a little bit. Note that if there are only red cards left in the deck, then it doesn't matter when we say to stop anymore - after all, all of the remaining choices are red! So we can choose to say "stop" immediately, or we can wait until the last card. Let's update the algorithm then as follows:

**Algorithm 2** SAFE BET 2: SAFER THIS TIME

```
procedure STRATEGY
    if There are only red cards left in the deck then
        Wait until the last card, then say "stop."
    else if There is only one card left then
        Say "stop."
    else
        Continue.
    end if
end procedure
```

Wait a second. In this case, our algorithm has just become "pick the last card"! So it turns out that our "Safe Bet" strategy was equivalent to picking the last card, which, while a bit disappointing, isn't all that surprising, if we believe that they're all equivalent anyway.

## 2.2 A Bolder Claim

You may be wondering why I spent half a page finding a second, less direct way to prove that the algorithm we chose has a 50% success rate. I know I am. But it does provide a neat segue into a new claim:

**Claim 3.** *Every legal strategy to play this game can be seen as choosing the last card in the deck.*

"But this claim follows directly from the previous one!" Yeah yeah, sure. But I feel that this one is more demoralizing: you spent all this time coming up with a hotshot strategy that you may or may not be proud of, and I'm going to show that all you're doing is choosing the last card in the deck. Isn't that fun?

*Proof.* The proof to this relies on a few sneaky propositions about the mechanisms of the game.

**Proposition 4.** *When you stay "stop," it doesn't matter which card I pick from the deck next.*

Your algorithm, whatever it is, has no information about the next card specifically, just about the makeup of the remaining cards in the deck. So if, say, I flip the second card from the top instead of the top card, it won't affect your algorithm's probability of success at all.

**Proposition 5.** *The bottom card will always be available as a card in the deck to flip over.*

That one is pretty obvious. Let's take a look at these two propositions combined! Together, they tell us the following:

**Proposition 6.** *When you stay "stop," your algorithm won't be affected if I were to flip over the bottom card instead of the top one.*

Can you see how this follows from the previous two propositions? But if that's the case, then no matter where your algorithm dictates to say "stop," I'll always be flipping over the bottom card! So the probability of you winning is exactly the probability of the bottom card being red, which, of course, is exactly 50%. □

## 3 Lessons Learned

There are so many reasons I love this problem, but I think the biggest reason is that it illustrates just how bad we are at reasoning with probabilities - I don't think anyone encounters this problem thinking it's even a possibility that there's no way to improve your odds. That said, I think it also carries some important lessons in algorithm design:

1. Always try to look at problems from different angles. When you're stuck on a problem, instead of trying to find a different solution, it can often be more helpful to revisit the problem itself. What would have happened in the case where the deck is only 4 cards? What if there were still 52 cards, but only 1 red card? 2 red cards? These alternate scenarios all provide hints to the central point of this problem.
2. Don't get carried away with optimality! We talk a lot about "optimal" algorithms, but in the real world, it's pretty often that the improvement over "good" algorithms is practically unnoticeable. If you tunnel too much on making the best possible decision at every point, there are always concerns of marginal returns, but you may also find yourself missing out on the bigger picture! Imagine how much less infuriating this problem would have been if I had asked "Is there a strategy better than 50%?" rather than "What's the optimal strategy" - by coming in with an inkling of doubt, it may have saved you a lot of headache.
3. Design analyzable algorithms. My example algorithms were very easy to analyze, but I suspect yours may not have been. If you design a reasonable algorithm that's easier to analyze, you may not only get insight into how well you're doing, but also where there are and aren't roadblocks to efficiency.