SRINI DEVADAS: All right, good morning, everyone. Oh, do we have a treat for you today. I'm going to publicize, for the first time, my hidden talent of mind reading.

And since this is a magic trick, you know, every magician needs an assistant. And I want to introduce Billy Moses to you. Billy is a PhD student. He already has three degrees from MIT, but he wants a fourth, I guess.

And so I'm going to turn it over to Billy, here. And Billy is going to sort of set things up. I'm just going to do my thing, which is read your minds, OK? So go ahead, Billy.

BILLY MOSES: So what's going to happen is, I'm going to-- I have this card deck right here, normal card deck, perfectly normal cards. They're not in any weird order. You can see them. I'm going to give five people the opportunity to pull a card and then give it back to me. They can look at the card. Then what's going to happen is, I'm going to turn over four of these cards, show you the final card. And then Srini is trying-- going to try to guess the fifth card.

SRINI DEVADAS: Well, but what I need you to do is, when I'm going to guess, I need all of you to really focus on that hidden card, I mean, in your heads, right? So I can't do individual mind reading, just in case you guys were worried. So if you think I'm wearing a terrible-colored shirt, I'm not going to know, OK? But if you all think of this hidden card, I'm going to be able to guess it. But so I need some help, all right?

BILLY MOSES: OK, anyone wants to try pulling some of the cards? OK.

SRINI DEVADAS: And I guess--

BILLY MOSES: Take your favorite card.

SRINI DEVADAS: --I can step out--

BILLY MOSES: --it can be anywhere.

SRINI DEVADAS: --here, right?

BILLY MOSES: Yeah.

SRINI DEVADAS: All right.

BILLY MOSES: And look at it. Give it back to me once you've-- cool. Anyone else want to pick a card? Want to pick a card? Here you go. You can look at the full deck, look at whatever card you want, pick it, cool.

SRINI DEVADAS: We're doing a little trick.

BILLY MOSES: Anyone else want to pick a card? You want to pick a card? OK, anyone else want to pick a card? Want to pick a card? Cool, want to pick a card? Cool, OK, I've got my cards all nice together.

OK, so five cards, I'm going to show you first. And then we're going to present them in a fun way for Srini to see. So the five cards are as follows. We have this wonderful Seven of Clubs. We have a Jack of Hearts. We have Seven of Hearts. We have Queen of Spades. And we have Ten of Clubs.

Now, for the people who pulled the card, these are indeed the cards that you gave me, right? No one missed their card in the deck? OK, I'll go get Srini.

SRINI DEVADAS: All right, ready for me?

BILLY MOSES: Yup. OK, the first card is the Seven of Clubs. The second card is the Jack of Hearts. The third card is the Seven of Hearts. The fourth card is-- and this is the final card that Srini will be able to see-- the Queen of Spades.

Now, everyone think to yourself. You've got to see the card, the fifth card, that Srini is going to have. Think it very closely in your mind, think.

SRINI DEVADAS: All right, definitely a clubs, definitely a clubs-- OK, see-- and I'm getting-- I'm getting at Ten of Clubs.

BILLY MOSES: And it is a Ten of Clubs.

SRINI DEVADAS: All right, thank you. All right, so I guess I'll have to confess. I'm not a mind reader, but I also did not cheat, OK?

So there's an algorithm here. And do you guys want to know what the algorithm is so you can go and try this out, I guess, when you go back home next, or whatever, at your dorms, or with your classmates? So it's not as easy as it might seem, simply because the combinations aren't really in our favor, right?

So if you think about it, there's 52 cards. So this is a regular deck of cards. You know, it's as I said, there's no cheating here. Billy showed me four cards.

So which means the fifth card could be 1 of 48, right? I mean, it's completely random. You guys picked these five cards completely at random, so there's no way that that fifth card could be anything specific, OK?

So what Billy had to do was, in the four cards that he showed me, he really had to communicate one of-- specifically, one of 48 possibilities, which is kind of-- you know, if you look-- if you think about it, then the math doesn't quite work out, right? And so if you just say-- so what are the ways-- how much information could you imagine communicating, in terms of bits of information, right? You know, in terms of possibilities or bits, how much information do you think that Billy could communicate in these four cards, which were also picked at random.

Obviously, they have to be specific cards, right? He just put them up there, right? So if I have four cards, what are the ways that Billy could communicate information. I mean, you saw what he did. Any ideas? Yeah, Ganatra?

AUDIENCE: The order the cards are placed.

SRINI DEVADAS: The order the cards are placed, so that's a great answer. But if you look at the order in which the cards are placed, how many different orderings are there? Four cards, how many different orderings?

4 factorial, right, and that is? 24, so I'm not quite at 48 yet, right? So that's a bit of a problem. So is there something else going on other than the ordering? Yeah?

AUDIENCE: The time rate at which he put them.

SRINI DEVADAS: So you mean the way he put them up, you know, slowly versus quickly? Or did you mean something with which was-- the time in terms of exposure of the cards, or?

AUDIENCE: Have some variations, for example, whether the red is before black.

SRINI DEVADAS: Well, but that's all part of the ordering. I mean, you have to be careful here, in that, I mean, you can't do both, right? I mean, you can't constrain the ordering and then say that there's 24 different orderings. The moment you constrain orderings, the number of orderings shrinks, right?

So I will say that, as I said, this is a very legit trick. It's completely algorithmic. We're going to write code for it-- or I'm going to show you code for it. But there's nothing else other than the ordering, OK?

He is-- there's no sleight of hand. There's no hidden communication. You know, he didn't stress words, you know, Clubs meant something, or you know, Spades meant something else. It was just the ordering, OK?

And so there's a gap here in terms of 24 and 48, right? And so what's cool about this trick is that we're actually going to be able to do this original trick, where the fifth card could be one of 48, because we're going to do one thing that you probably didn't notice, because we only did the trick once, all right? And so let me explain how this is going to work, with just sort of randomly picking-- I'm just going pick the top 5 cards, assuming these are shuffled. And you can assume these things that are at random, right?

So I have five cards that I've picked here. And can you all see them? OK, what do you notice about these five cards? Anything-- yup, Fadi.

AUDIENCE: There are four types and five cards so there's at least one that is repeated.

SRINI DEVADAS: Right, so the-- so you're saying the suit, right? So it is clear. This is true here. You have the Seven of Clubs and the Ten of Clubs, right? Oh are these the cards that you pulled out, or close? Anyway, so they look pretty similar to the ones we used, but that's probably just because we didn't shuffle the deck.

But obviously, if you have five cards, and you only have four suits, I guess you could call it the pigeonhole principle from 6.042, but you're definitely going to have a pair of cards that are going to be of the same suit, right? I mean, there may be more than two cards of the same suit, but you're guaranteed, for any five cards-- let me just pick another five-- that two of them are going to be of the same suit, right? That's pretty much guaranteed. And so we ended up getting clubs again here. And so you have Two and Four of the same suit.

So the first thing that happens in this trick is, we use the first card to give away the suit. Or Billy used the first card to give away the suit. Doesn't quite finish the story here, because let's just say that I'm going to use the first card to signify the suit. So in this case, let's just say I am going to do Clubs.

Now, I have a choice here. I could-- Billy could have hidden Two, and showed me the Four of Clubs as the first card, right, or vice versa. And keep that in mind, because that's actually incredibly important. But if you just say that the first card is going to signify the suit, then let's say the suit is now out of the picture, so I know the suit of the card.

How many-- so I have three cards left, all right? And I got a hidden card. I know the suit of the hidden card, but that card could be-- the hidden card-- how many possibilities are there for the hidden card now? How many possibilities? I mean, this is all at random. Well, 13.

AUDIENCE: 12

SRINI DEVADAS: Oh, good point, good, good, good-- see, this is exactly why I need attentive students. OK, good, so there's 12 possibilities since one of them was-- one of them was shown, right? So did that help us? There are three cards left, and so how many different-- I mean, the first card is the first card, so it's no longer part of this permutation.

I can't count it in terms of the permutations. There's only three cards left. And those-- certainly those three cards could be ordered in a certain way, but how many combinations or permutations are there of three cards?

Only six permutations, so again, I seem to have a problem here. I don't seem to have solved the original problem of the 24 to 48. I still have a 6-12 gap, right? So we're going to have to somehow cover that gap.

And any thoughts? I gave you a little bit of a hint as to how this gap is going to be covered when I talked about the Two and the Four. How am I going to cover this gap, the 6-12 gap? Yeah, Kanishka.

AUDIENCE: Possibly by saying odd or even. So if the next card you can say odd or even you shrink it down.

SRINI DEVADAS: But you mean-- say, you mean Billy is going to say something?

AUDIENCE: No, he's going to present the next card.

SRINI DEVADAS: Yeah.

AUDIENCE: Whether it is thinking that the final card you have to guess is odd or even and you can shrink down the possibilities.

SRINI DEVADAS: So, I mean, he-- the only thing he can do is give me an ordering of the remaining three cards, right? Yeah, go ahead. Kevin, right?

AUDIENCE: Yeah, maybe you used the cards to make a low-to-high limit on the range.

SRINI DEVADAS: That's good, and low-a-high limit on the range, exactly right. So what's happening here is, not only do I have the first card-- and it's not the first card is just giving me the suit. I mean, that's the way I said it, right?

But the first card also has a number on it, right? The first card has a number on it. So that number could be an anchor that is going to help me figure out what the hidden card is, right?

And what's important to understand, and this is really the last interesting point with respect to shrinking these possibilities or these permutations down, is you want to think about your-- let me draw this out properly in a nice little circle. You don't want to think about just a ascending sequence of cards or numbers. You want to think about this as being circular, as you'll see, for the following reason.

All right, so let's just say the ace is a 1. And so we're going to think of that as being the smallest number, so you go 1 through 13, in this case. The king is a 13. All right, so that's your number. And so you could imagine you'd go 1 through 13.

So let's just do that first. And then I'll explain to you why I drew it in a circle. And the first thing that happens is, I'm going to take some card here. And I'm going to make that the first card, so I don't care about suits anymore.

And if I just picked this-- another five cards. Let's just do this again, but with another set of five cards. And what do I have here? Oh, I've got mostly hearts, OK. So let me do something more interesting. This is too many hearts here.

Right, let's do this. So now, I got a Two and a Jack. And I got a Ten an an Eight. OK, now I'll make this even more interesting by picking-- I want to pick-- I want to get a hearts card. And let's see how this works.

Ah finally, all right, so this is what I have, King of Hearts, the Eight of Spades, et cetera. So the two cards with the same suit are the Two and the Jack. And so here's the important thing. One of the things that Billy did, kind of, in a subtle way, I'm assuming, and some of you may have picked up on it.

The very first card that he asked one of you to pick certainly wasn't the hidden card, right? I mean, he got all five cards. And then he decided which of the cards was going to be the hidden card and then took the remaining four cards and encoded them.

And if you don't do that, if you just say this, if you-- this trick won't work if this is the hidden card. This trick is only going to work as it turns out if the hidden card is a Two. It won't even work if the hidden card is a Jack.

And there's a reason for that. And you-- can you think of the reason for that if you look at the board? Can you think of the reason for that? Someone other than Fadi? Someone else?

What do you see up here? So what am I trying to do? In terms of-- what would happen-- you guys talked about-- Kevin mentioned distance and things like that, right? You know, you call it the offset.

So what would happen if I hid the Jack, and I did this-- I did something like this, where I-- the Two of Diamonds was the first card that was put out. And then I tried to encode and number using these remaining three cards, what's the biggest number that I could encode? I mean, in-- I need to encode lots of numbers, right? So what's the biggest number that I could encode here?

If I had six permutations, and I start with one, the biggest number that I could encode is a 6, right? Sadly, the Jack is too far away from the two, right? If I think of the Jack as being an 11, it's 9 away from the 2, right? But help me out here, what could I do? If I flipped it, what happens?

AUDIENCE: Count backwards

SRINI DEVADAS: You could count backwards. Or what do I have up there on the board? What do you see up there on the board? I wrote something that was linear, but what do I have up there on the left? Circular, so you think mod, mod arithmetic you know, think of this as being circular, OK?

So what happens is that, if I picked a 2 and a Jack, if I-- what I want to do is to go clockwise. OK and because there are only 13 cards, for any pair of cards, it's always the case that I can choose the first card in such a way that the hidden card is, at most, 6 away from the first card, OK? Because you could get into a situation where, obviously, if I ended up hiding the Jack and exposing the 2, I end up getting a 9, which is impossible to encode. But if I hide the 2 and expose the Jack, then I go 1, 2, 3, 4, and I am at, basically, 4, OK? So that's the last thing.

So let's just do this really quickly with yet another set of five cards. Now obviously we practiced so we could do this fairly quickly, right? But-- and you can practice too, but sometimes with things you get lucky, you know, and you get easy things to encode.

So what happens here? Someone wants to-- so now that you know the algorithm, you have some choices here, but obviously you have-- not all choices would work. You've got to be a little bit careful. What would you hide?

So you need to make sure that you get-- you can choose a pair of-- I'll make this a little bit easier. Sometimes choices make things more difficult. Perfect, all right, so what you see here is-- you know that you have to do the clubs now, because you have the clubs as being the suit that's repeated. No other suits are repeated, right?

So what am I going to expose? And what am I going to hide? That's your first question, right? Someone else?

You guys are all going to have to do this trick. That's the final exam for this class. On Friday next week, you're all doing the trick-- no. So what am I going to-- yeah, go ahead, Josh?

AUDIENCE: You are going to hide the Ace, the King of Clubs would be the first card.

SRINI DEVADAS: You're going to hide the Ace, exactly right. You want to hide the Ace, because you want to go this way. And if you do it the other way around, it's not going to work. You're going to hide the Ace. And you're going to make the King the first card.

And now, what you need to do is-- this is gone, right? And now what I need to do is, I need to encode a number between 1 and 6, in general. In this case, I need to encode the number 1 in order to give away the fact that the hidden card is an ace.

And now I could do what was suggested before, which is have some sort of-- I mean, there's obviously an ordering associated with these cards that you see up here. If I call this a 1 and I call this a 13, then lets do the logical thing, which is, 1 is smaller than 2, which is smaller than 13, but I might end up-- I might have and Nine of Hearts here rather than a Five of Hearts. And I got to, obviously, be able to order the Nine of Hearts in relation to the Five of Hearts as well, so-- because I-- all of these could be nines actually, right?

And so I can't just rely on the number to do the ordering, so I can just choose an arbitrary-- clubs is smallest. Diamonds is the next. And then I got Hearts. I just did this because this is a little bit easy to remember for me. You can do this in many different ways, in alphabetical order, you know, C-D-H-S.

And so this-- that's moot here. What I need to do is to encode the number one, OK? But I encoded-- in order to encode the number one, I have to have some sort of way of ordering this. And so what I'm going to do is, I'm going to say SLM is one.

So what do I mean by SLM? The smallest card is first-- I'm sorry, SML is-- increasing. The smallest card is the first one. The medium card is the second one. And the largest card is the third one, OK? And I'll write this out.

And you can see the code for it. But when you see something like LMS, you're going to get a six. And there's obviously six of these different combinations, right? The details are actually entirely up to you. I mean, the most important thing here is the ordering that Billy had had to be ordering that I had.

And so yesterday when we practiced, one time, there was a mismatch. And we got it wrong. But that's the only way the trick would fail, really, I mean, other than human error, is if the-- is the encoding. This is really encoding followed by decoding.

And obviously, you need the same algorithm, if you will, which requires that you need the same ordering, in this case, for the cards, right? So that's it. That's the trick. It doesn't seem that interesting anymore, huh?

But no, it is interesting if you show it to people. And the fun part is in actually doing it. So I encourage you all to do it. Any questions about how this trick works?

So what I'd like to do is show you some code for this trick. And there's also a code that was useful to both of us, because we could practice by ourselves. We didn't have to be in the same room to practice, because I could randomly generate a bunch of these four cards. And obviously, the program knew what the hidden card was.

And I could-- I'll show you the code. I could type in what the hidden card was. And it would say-- you know, I guess it would say, you're a mind reader extraordinaire, or sorry, not impressed. And Billy could do the same thing in terms of checking his encoding, all right?

And then, after we're done here talking about the code-- and there are some interesting things with the code-- let's talk about whether you could-- talk about different kinds of tricks. And in particular, I want to talk about whether we could go from five cards to four cards using a different encoding scheme. Cool, so I wrote this code about a year ago. And I think I'd probably write it differently now, but we have what we have.

So the first thing is you have to decide on an order. You absolutely have to decide an order. And the cards themselves, as I mentioned, there's a natural order associated with the numbers. And of course, you can decide if you want to start with a two, and then the ace could be 14-- I mean, kind of makes sense for you to go 1 through 13.

So that's really what we have here. The ace is the smallest card. And the King is the biggest card. And as I mentioned, the clubs are smaller than the diamonds, et cetera.

So I just have a deck of cards that is ordered as you see up there. And that's just a Python list that tells you what the ordering of these 52 cards are. Does that make sense?

So there's many procedures here, many routines that do different things. And a lot of them use the same code, so I'm not going explain each one of them. But this one is a good one.

And I'll also show you the computer assistant one, which is essentially this procedure except that it allows-- allowed me to practice by myself and generated cards randomly. In this case, what happens is, you input five cards. And what this does is-- so you could sort of take five cards out from the deck. And then manually, you can choose the hidden card, order the remaining four.

And if you typed in those five cards into this particular procedure, it would give you the first four cards, OK? And so you can then verify that you've done the ordering. So this would be something that-- I don't know if Billy used this to practice, but he could have.

So what do I have here? Well, the first thing is just printing out what the order is. There's a bunch of things that need to get initialized. Cards is simply a list of five cards.

And then you have c index, or cind, which is the indices in that giant list that we have that had 52 cards in it. They're just the indices of the various cards. You can use .index, which is a method to go from a card, which is like a_c, for example, to the index.

And then card suits are what suits are for the cards. And it's 0, 1, 2, and 3 for clubs, diamonds, et cetera. And then the numbers are what the numbers of the cards are, 1 through 13, OK?

You kind of need-- there's a bunch of bookkeeping. As you can imagine, we're doing things in our head here, but if you want to get a computer program to actually do all of these computations, it's not complicated, but you have to worry about the four suits. And you've got to worry about the 13 cards.

And you have to go ahead and encode SML, or produce this encoding, right? So you can imagine there's a bunch of code. It's not complicated code, but there's a bunch of code.

So the first thing that happens here is, in this particular procedure, you want to input put these cards in. And so let me just run it. And so you get a sense of what this procedure does. All right, beautiful. So let's just say-- let me just pick five of these here.

So the first one is Five of Hearts-- I'm sorry, Five of Hearts, yup, Five of Hearts, Nine of Diamonds, Jack of Spades, Ten of Hearts, Eight of Hearts. And so it essentially says the first-- so if I have something like this, it ended up choosing the Ten as the hidden card, and decided to choose Five as the first one.

And then because Ten was the hidden card, you needed to encode 5, right? And so the way you encode 5 is by encoding L S and M. So the first card that should be up is a Jack, followed by the smallest card, which is an Eight, followed by a Nine. So that's essentially it.

So now that you've seen that, let's take a look. I'm not going to go through every line of code here, but I want to point out a couple of things that are interesting. This is simply filling in the data structures.

The only thing that's interesting here, or the thing that's probably the most interesting here is what I've highlighted. This is a way of taking an element in a Python list and getting the index for it. And so that's essentially something that-- it's object-oriented programming. We probably want to get into that in this class, but you'll definitely see that in 009 and other classes.

And essentially, what we have here, given the ordering, is the suits simply correspond to whatever the indexes mod 4. Because, as you can see from this, the clubs are-- this is 0, the index 0, and the club is index 0, 1, 2, 3, 4. So you're going to get mod of clubs is always going to be-- mod 4 of clubs is always going to be a 0. Mod 4 of diamonds is always going to be a 1, et cetera, so that's essentially what happens here.

And the first thing I need to do is figure out two a card that have the same suit, because that's important, because that's going to determine what my first card is and what my hidden cards are. And that's what happens over here. And then once I have that, I need-- I've just got to split these five cards into two piles, one of which has the hidden and the first card in it, and the other pile has the three cards that correspond to the ordering in it.

Obviously, the computations that I have to do for the first couple are quite different from the ones for the remaining three cards. In one case, it's ordering. In the other case, it's a comparison.

And I do have to-- the most interesting remaining part of the code that I'm going to show you is the determination of what the hidden card is and what the first card is. And that's going require-- this is the part that actually took me the longest to code. I got it wrong a couple of times. It's going to require me-- it's five lines of code, but it's subtle. It's going to require me to kind of figure out whether it's a jack versus two, you know, or vice versa.

And things get a little bit interesting. For example, if you had-- if it wasn't the Jack and the Two, the example I had here, so let's do this-- put the Jack and Two. Let's say I had a Three and a Ten, right? They're kind of the furthest apart.

And you had to be kind of super careful here, because 3 to 10 is obviously 7, which is too large, but you need to go 10 to 3, which is 6, right? So the things that are diametrically opposite to each other are usually the hardest when it comes to actually doing the trick, but it's also hard from a standpoint of coding. I had to get the mods right, and the subtraction, and in terms of what got subtracted from the other.

So the rest of it is-- I just have high level descriptions. Obviously, the code for these individual subroutines is in the file as well, but this one determines what's hidden, what the other card is, and what the number that you need to encode is. So when you go do the split, and you have the two cards, you obviously need to decide which one goes first, which one is hidden.

And you also get the number. And that number had better be between 1 and 6, otherwise this is not going to work. And then you remove the other-- the two cards hidden in other from the set of five cards. In order to actually do this SML, it was a little easier for me to sort these cards.

And then, depending on the number and code, I'd change the order. Obviously, if I sort the cards, I'm going to get SML. And that may not be what I want, but it's easier to start with SML, saying, this is the order that gave me the number 1 as an encoding. What do I need to do in order to change that encoding to a 2, or 5, or a 6, or what have you? OK, and so that's what this outputNext3Cards does.

This is what took me the longest to code. I mean, it was a few minutes, but the rest of it was pretty much busywork. There wasn't too much thinking. So this thing here, as you can see, the encode is done-- you don't need to worry too much about the list itself, and the indexing into it.

The bottom line is I need to figure out this situation. And I need to subtract. And so when I see something that says 10 and a 3, I end up getting a 7. And then I know that I have to do a flip, OK? But it also may be the case that I might get something that-- the reason I need the mod 13 is partly because I might end up getting a negative number.

So it's not clear that encode, take away the mod 13, is going to give me a positive number. I might be subtracting 9 from 2, so I end up getting a minus 7. So you don't know this. I mean, the program doesn't know which one is which. And it has to figure out which one is which and flip them together.

So if there's any code that you want to look at carefully, it's this code. And it's just a matter of sort of going through the combinations and making sure. I know it works, but you need to convince yourself it works. So that's pretty much it. There's not much more to say here that wouldn't be obvious to you looking at it carefully, so I don't want to belabor the obvious.

And outputNext3Cards is the part that I just described. You start with the default ordering, which encodes a 1. And then you need to do a little bit of flopping, if you want to encode a 3, for example.

And if you want to encode a 3, then you'd change things around. And this is exactly what I described to you before. So this is actually fairly straightforward. If you have that table written in front of you, you just go off it and write that.

Certainly if you change the algorithm, all of this would change quite dramatically. And there's an exercise which asks you to do that, which you could certainly try to do. I want to show you one other thing, which is interesting. And then I want to run this new subroutine.

So what I have here is, I'm going to run-- oh, I've already run it. So I want to show you computer assistant. It was very useful to me. And this says, please give me a random number of at least six digits.

So obviously, I don't want to know what the hidden card is if I want to practice in a legitimate way. So what I did was, I said, I'm going to have this sort of hokey-- I mean, it's really hokey. I'm going to have this hokey algorithm that's going to take this giant number that, I'm going to type in.

And it's somehow going to take this giant number. And it's going to deterministically generate 5 cards based on this giant number, but I didn't want to have an obvious way. I mean, I don't want to say the number 12345 and then end up getting the first five cards in the deck, or something like that, because that would sort of give it away right. So I want to have some sort of obfuscated way of taking a giant number and generating something that was at least unpredictable to me in terms of the five cards that are generated.

And so what I do is, if I type in a number, some random number. And at this point, I have no idea what the five cards that are going to be generated are. And this just tell me-- it does do the assistant order the cards functionality, so it does take those five cards, hides a card, and then exposes the four cards to me. And so I'd get different-- depending on what number I type.

So the way I'd practice is I'd just sort of close my eyes and just type some random number. And then it would produce four cards for me. And notice that this is wonderful.

Notice that there's a bug, right? This is exactly why I put this up. And wow what luck, so notice that this didn't work.

You know, and that's the fix. That's an exercise. I'm not-- I kid you not. It's an exercise that's going to be up there, which is fix my hokey algorithm so it doesn't do this.

And so what happened here was, I had a deterministic way. This wasn't a random number generator. The good way of doing this would have been to use Python's random number generator facility and not have me do anything, because obviously I attached something that turned into that, which is a deck that has two King of Hearts in it.

And so my deterministic way, which was a convenient way because it's only four lines of code didn't need me to include the Python random library, et cetera, causes trouble. So this obviously won't work. So I'll just type something. And it's going to say, sorry, not impressed.

OK, not impressed with my guess and not impressed with my code, so let me do this again. And so let me just do that. And now-- what? Is this-- this is ridiculous. I mean, does my code even work? I'm going to blame it on my computer. I told you my laptop is acting up. OK.

1111222-- ah-- thank you, all right, so now I see this. And I say Seven of Clubs, so that's the first card. So clearly, it's a clubs. And then Ten, King and Six, so Ten is medium. So I got medium, large-- MLS, right?

So MLS, is-- MLS would be 4, right? So 4 would give me the Jack of Clubs, right? There you go. So this worked, OK?

But now I want to show-- you the last thing I'll do is show you this code. And you guys should absolutely fix this code. It is clear what your exercise is. Or you could-- I do want to talk about doing this with four cards.

So, as note is that, this is the offending line of code, OK? So what I did was, I wanted to generate five, kind of, random-looking numbers out of this giant number that I typed in, right? And I got lazy. And I was tired of this puzzle by the time I ended up coding it.

So I just said, you know, I want-- I don't want to include random.lib from Python. I'm just going to take this. And I'm going to do some sort of iterative computation, where I'm generating what I think are random numbers between 1 and 52 from this giant number that I typed in.

And I'm going to do this five times. And what happened, as you saw, is that I generated the same number twice a couple of times, right? And that's why, obviously, the trick doesn't work in that case, right?

What was interesting was, the way I found this bug was not the obvious way that we just saw here. It was actually something where the hidden card was the same as one of the four cards that was exposed. And I didn't know that, right?

So I'm like, what? Why am I not getting this right? It kept saying, sorry, not impressed. And what it meant was, it wasn't impressed with my coding, not with my deductive skills.

So it took me a while to figure this out. And you can fix it. OK, so that's it. Any questions about the-- what we've talked about, the puzzle, or the code?

OK, so in the five minutes that we have left, let's talk about whether there would be a trick which would be perhaps even more impressive where we would only use four cards. So it would only be four cards that the audience selects. And we would-- Billy, I guess, would show me three cards. And I'd have to guess the fourth.

Would that be possible? I mean, the numbers really don't work right, because that fourth card could be one out of-- I guess, if you show three cards, one of the 49 different possibilities. And it would be pretty difficult, right? You could think about something where-- the only way I could come up with was-- it required much more encoding, but I was thinking about something like this, where let's say I have these four cards, right?

And let's even take-- let's say that they're all of different suits, right, because that's possible. The other thing that happens when you have-- when you only have four cards is they all four could be of different suits, right? That's certainly possible.

What do I want? I want a heart. OK, good, so I have something like this. And now I'm in real trouble. I mean, clearly, anything that we talked about isn't going to work. So let's just say that I end up choosing one of these cards to hide, right? So this Ace is hidden, because it's not quite clear what I can do here, but bear with me for just a minute.

So for argument's sake, let's say that now, at this point, I'm trying to get more information out of this. The ordering isn't going to help me. I mean, the ordering is going to help me a little bit, but what am I going to get, like 6-- 1 through 6 from this?

So I could say the first card is an anchor, OK? And so I'm going to expose this. So the only thing I could think of was, you could sort of do-- this is a little bit obvious. And I'm maybe giving things away.

But depending on where you have the hidden card, and you can get more information based on the order in which you put these cards out and whether they're to the left or the right of this hidden card. So you'd need to do something like that. I mean, I haven't figured this out. At least, I can't think of a complete solution to this puzzle, but I do feel like that you can get some information, additional information based on whether you put things to the left or the right.

And that's clearly a bit of information that you get. And maybe you could add to that. And you know, think about it. I think it makes for an interesting exercise.

And obviously, there's other ways of doing the five-card trick, as well. Cool, all right, that's all I had for today. If there are no questions, we can close this particular session.