

USING A CARD TRICK TO TEACH DISCRETE MATHEMATICS

SHAI SIMONSON AND TARA S. HOLM

ABSTRACT. We present a card trick that can be used to review or teach a variety of topics in discrete mathematics. We address many subjects, including permutations, combinations, functions, graphs, depth first search, the pigeonhole principle, greedy algorithms, and concepts from number theory. Moreover, the trick motivates the use of computers in mathematical research. The ultimate solution to the card trick makes use of Hall's Distinct Representative Theorem.

1. INTRODUCTION

An interesting card trick is presented while telling the story of how our discrete mathematics class analyzed, attacked and solved some of its mysteries. The trick is a model for engaging students in mathematical research using computers and programming as tools. The presentation is based loosely on what actually transpired in class. We discover theorems, refute conjectures, verify others, and leave work for the future. The story has a recurring theme of making progress just when it seems that the options have been exhausted.

1.1. **The Trick.** The original trick appeared in 1950 in *Math Miracles* [L]. It was invented by Fitch Cheney. It was largely ignored until 1986, when Art Benjamin showed this trick at the Hampshire College Summer Studies in Mathematics. Several mathematicians have since analyzed it, discovered strategies, and performed it around the country [K2], [M1], [M2].

The trick makes use of a variety of combinatorial ideas, including a neat application of Hall's Theorem (see, for example, [CL], [BM], or almost any introductory graph theory text). It provides a good review for the many concepts discussed in a discrete mathematics course. In addition, the mathematics behind the trick is a natural candidate for experimentation using computers. The trick is also fun and entertaining on its own.

1.2. **The Class.** The students involved in this report were from ArsDigita University (ADU). ArsDigita University was an experiment in computer science education. ADU offered bright and motivated college graduates an undergraduate level major in computer science in one intensive year. The students were at the school 10+ hours a day attending lectures and recitations, and working on problem sets. Each semester course was compressed into four six-day weeks. The students' efforts on this card trick took place over the last week of the month.

Admission to ADU was extremely competitive, and 50 students were admitted from some 400 applications. The background of the students included English, History, Psychology, Law, Medicine, Engineering and every likely major except computer science. What the students lacked in mathematical background, they made up for in maturity, experience, and effort. ADU offered a quality (MIT caliber) undergraduate computer science education, free of charge, preparing its graduates for jobs in the industry or graduate school in computer science. More information about ADU can be found at [Si1] and <http://aduni.org>.

In other words, we had a great audience, and the report here must be taken in context. Some of the things that worked with our group may not work with a typical undergraduate class at the sophomore level. However, a similar lab based on this card trick was successfully classroom tested at Stonehill College, in a "learning community" course combining mathematics and computer science. The students at Stonehill were typical college sophomores majoring in either mathematics or computer science. That lab can be found at http://www.stonehill.edu/compsci/LC/card_trick.html.

2. THE CARD TRICK

A professor and teaching assistant (TA) enter class and announce that they will begin class today by performing a magic trick. The professor turns his back to the class, as the TA offers a deck of cards to a student and instructs her to choose any five cards randomly from the deck. The cards are the $4\heartsuit$, $7\heartsuit$, $9\spadesuit$, $J\clubsuit$, and $9\heartsuit$. The five cards are held up so that the TA and everyone except the professor can see them. The TA takes the five cards and gives one back to the student telling her to show it to everyone and put it away. The student holds up the $7\heartsuit$, and puts it in her pocket. The professor is finally invited to turn

around, whereupon the TA in a grand exaggerated display places the four cards, one by one, on a table for everyone, including the professor, to see. The cards in order are $4\heartsuit$, $9\spadesuit$, $J\clubsuit$, and $9\heartsuit$. The professor watches her, looks at the students, looks up at the sky, and after an appropriate suspenseful patter, correctly announces the card that the student has put away, the $7\diamondsuit$.

2.1. Is it Magic or Mathematics? In the discussion that follows, the reader should note that the students' first suggestions are not always correct, but that the suggestions eventually evolve and lead to a correct analysis.

One skeptical student claims that there must be some trick. There are 48 cards that are hidden, and the professor needs to choose one of these 48, but the TA can only convey 24 pieces of information, because there are only $4! = 24$ ways to order the four cards she shows.

A number of students suggest that the TA is using some method of conveying more than 24 pieces of information. There might be some trick in how the cards are oriented, or read out to the professor. In order to test this hypothesis, the class demands to see the trick again, but this time the professor must leave the room, the TA must neatly stack the four cards in a pile face down all in the same direction and leave the room before the professor returns, no talking is allowed, and a student is designated to hand the pile of four cards to the professor.

We (Professor and TA) follow the demands precisely and the trick still succeeds. When the class asks to shuffle the four cards before the professor sees them, we say that this will cause the trick to fail. The TA insists on both choosing and ordering the four cards herself.

The skeptical student now realizes his error. There are more than 24 pieces of information, precisely because the TA can choose which four cards to show and which to hide. There are five ways to choose which card to hide, and there are 24 ways to order the four remaining cards. This gives 120 possibilities altogether, and plenty of room for the TA to relay the necessary information to the professor. Thus, the trick can in principle be done with a deck containing as many as 124 cards. The *maximal deck size* for a given n is the maximum number of cards d such that the number of unseen cards ($d - (n - 1)$) is at most the number

of pieces of information that can be conveyed by the $n - 1$ cards that are shown. Thus, for $n = 5$, the maximal deck size is $d = 124$. The maximal deck size then is an upper bound on the size of the deck for which our trick still works.

Theoretically, the trick for $n = 5$ can be done with a deck of up to 124 cards; however, how it is actually accomplished with even 52 cards is, at this point, still a mystery to the students.

2.2. Upper and Lower Bounds. The students decide to consider simpler cases of this trick, trying to determine exactly how to do the trick for a given size deck. Suppose we only choose three cards from a deck rather than five. Applying our previous analysis, there are three ways for the TA to choose two cards from the set of three, and two ways to order each of the two card sets. This allows us to identify a maximum of six cards. Adding the two visible cards gives a maximal deck size of eight. The class is not able to find a way to do the three card trick with a deck of eight cards. However, they are easily able to do the three card trick with a deck of four cards.

Suppose we choose three cards from a four card deck. The TA shows the professor two cards. There are $2!$ ways to order the two cards shown by the TA. We make the low-high permutation represent the lower of the two remaining numbers, and the high-low permutation represent the higher of the two remaining numbers. For example, order the cards one through four, and imagine that the TA shows two and four in order. Then the professor knows that the missing card is the smaller of the two remaining cards one and three, namely one. This implies that the trick can be done with a deck of four cards, when we choose three of them. We will call a number d the *lower bound* on the deck size if we have a method for performing the trick, choosing n cards from a deck of size d . Using this terminology, our current lower bound for $n = 3$ is $d = 4$.

The maximal deck size described in § 2.1 and the method just described in the last paragraph can be generalized to get an upper bound and a lower bound, respectively on the deck size for the n -card trick. The upper bound is n (the number of ways to choose $n - 1$ cards from n), times the number of permutations of $n - 1$, plus the number of visible cards, $n - 1$. This equals $n(n - 1)! + n - 1 = n! + n - 1$. Currently, our best lower bound is

the number of permutations of $n - 1$, plus the number of visible cards, $n - 1$. This equals $(n - 1)! + n - 1$. Our bounds are summarized in the table below.

Number of Cards Chosen	Lower Bound on Deck Size d	Upper Bound on Deck Size d
$n = 3$	4	8
$n = 4$	9	27
$n = 5$	28	124
n	$(n - 1)! + n - 1$	$n! + n - 1$

A student, who until now listened quietly and doodled, announces that she can push the lower bound up to five, for $n = 3$. She showed us the table below.

123	124	125	134	135	145	234	235	245	345
1 - 2	2 - 1	2 - 5	3 - 4	3 - 5	4 - 5	2 - 3	3 - 2	2 - 4	4 - 3

The top row lists all ten possible ways to choose a set of three cards from a set of five. Below each one is the ordered pair of cards chosen from this set of three cards. The TA and the professor use this table to both code and decode their communication. For example, if the three cards chosen are 4, 5 and 2, then the TA shows 2 and 4 in that order. When the professor sees 2-4, he knows that the original three cards were 2, 4 and 5, hence the missing card is 5. Unfortunately, the students realize, this strategy is neither easily generalized, nor easily computable. The easiest way to use it is to memorize it.

2.3. What is a Strategy? A strategy is a table like the one above where each ordered set in the bottom row occurs at most once. Let d be the size of the deck of cards, and n the number of cards we choose from the deck. Then a strategy is a one-one function

$$\left\{ \begin{array}{l} \text{Unordered subsets of} \\ \text{size } n \text{ from a set of size } d \end{array} \right\} \xrightarrow{1-1} \left\{ \begin{array}{l} \text{Ordered subsets of size} \\ n - 1 \text{ from a set of size } d \end{array} \right\}.$$

We work together as a class to try and determine a strategy for $n = 3$ and $d = 7$. We succeed by trial and error in a completely ad-hoc way. The successful result is shown below.

123	124	125	126	127	134	135
1-2	1-4	1-5	1-6	1-7	1-3	3-1
136	137	145	146	147	156	157
3-6	7-1	4-5	4-6	4-7	5-6	5-7
167	234	235	236	237	245	246
6-7	2-3	3-2	2-6	2-7	2-5	6-2
247	256	257	267	345	346	347
2-4	5-2	7-5	7-6	3-4	4-3	3-7
356	357	367	456	457	467	567
3-5	5-3	6-3	5-4	7-4	6-4	6-5

This moves the lower bound for $n = 3$, from five to seven.

2.4. Can we reach the upper bound? The class wonders whether we can move the lower bound upwards for other values of n . Can we move it up with a simply computed strategy instead of one that needed to be memorized? Can we ever achieve the upper bound, simple strategy or not?

Attempts at moving up to a deck of size 8 for $n = 3$, proves difficult. We often hit dead ends and have to backtrack. We feel there are good chances to succeed, because not all the ordered pairs appear in our strategy for $d = 7$. For example, the strategy above does not use the pairs 2-1, 4-1, 4-2, 5-1, 6-1, 7-2, 7-3. One attempt at building inductively on our ad-hoc method for $d = 7$ is shown below. We use all the old values appearing in the case for $d = 7$, inserting new values under a triple with an 8 in it, by first trying in order, all the unused pairs listed above, and then trying the pairs with an 8 in them in lexicographic order. Here is the failed result.

123	124	125	126	127	128	134
1 - 2	1 - 4	1 - 5	1 - 6	1 - 7	2 - 1	1 - 3
135	136	137	138	145	146	147
3 - 1	3 - 6	7 - 1	3 - 8	4 - 5	4 - 6	4 - 7
148	156	157	158	167	168	178
4 - 1	5 - 6	5 - 7	5 - 1	6 - 7	6 - 1	7 - 8
234	235	236	237	238	245	246
2 - 3	3 - 2	2 - 6	2 - 7	2 - 8	2 - 5	6 - 2
247	248	256	257	258	267	268
2 - 4	4 - 2	5 - 2	7 - 5	5 - 8	7 - 6	6 - 8
278	345	346	347	348	356	357
8 - 7	3 - 4	4 - 3	3 - 7	8 - 3	3 - 5	5 - 3
358	367	368	378	456	457	458
8 - 5	6 - 3	8 - 6	7 - 3	5 - 4	7 - 4	4 - 8
467	468	478	567	568	578	678
6 - 4	8 - 4	Stuck	6 - 5

When we get stuck at 478, we need to backtrack and change some old entries, in order to free up some possibilities. If there is no possible way to fill in this table, we will eventually backtrack and try every single possible table before we stop and report that there are no successful strategies.

Some of the hackers in the class start paying attention. One suggests that we use a computer to search for an appropriate strategy. “How difficult could the search be for a deck of size 8 and $n = 3$?”, he wonders aloud. “At least then we will know one way or the other whether we can achieve the maximal deck size for $n = 3$.” The class agrees that if we have no better plan, then this will at least give us some results. While the hackers write the program, the theoreticians compute the worst-case scenario.

If d is the size of the deck, and we choose n cards from the deck, then there are $\binom{d}{n}$ slots in our table, and each slot must hold an ordered subset of size $n - 1$. Therefore, the number of different tables for a given n and d is exactly $(n!) \binom{d}{n}$. That is quite a large number. Even if we assume that each table could be generated by a computer and checked for duplicates in one billionth of a second, then for $n = 3$ and $d = 8$, we get 6^{56} tables. Dividing by a billion we find that the program could run for approximately 3.77×10^{34} seconds, or 1.195×10^{25} centuries. This is not a promising approach.

A sensible person from the back of the class suggests that we analyze the simplest case, when $n = 2$. Maybe we can easily construct a strategy for the upper bound by hand in this case. The maximal deck size for $n = 2$ is $2! + 1 = 3$. In this version of the trick there are only three cards. A strategy is easily computed. In fact, there are exactly two strategies.

12	13	23	and	12	13	23
1	3	2		2	1	3

For the first strategy, if the professor sees 1 he knows the hidden card is 2, if 3 then 1, and if 2 then 3. It is even easy to memorize. If he sees x then the hidden card is $x + 1 \pmod 3$. The second strategy has a similar description. Thus, for $n = 2$, we see that the upper bound on deck size is the same as the lower bound.

2.5. A working computer model. At this point, one student's program finds a strategy and displays the following solution.

123	124	125	126	127	128	134
2-1	1-2	5-1	6-1	7-1	8-1	3-1
135	136	137	138	145	146	147
1-3	1-6	1-7	1-8	4-1	1-4	7-4
148	156	157	158	167	168	178
8-4	1-5	7-5	8-5	7-6	8-6	8-7
234	235	236	237	238	245	246
3-2	2-3	6-2	7-2	8-2	4-2	2-4
247	248	256	257	258	267	268
2-7	2-8	5-2	2-5	5-8	2-6	6-8
278	345	346	347	348	356	357
7-8	4-3	3-4	7-3	8-3	5-3	3-5
358	367	368	378	456	457	458
3-8	6-3	3-6	3-7	5-4	4-5	4-8
467	468	478	567	568	578	678
6-4	4-6	4-7	6-5	5-6	5-7	6-7

Here is how the program works. It uses a *greedy algorithm* to fill each entry of the table, one at a time. For every entry labeled abc , it generates the six possible permutations in reverse lexicographic order, namely cba , cab , bca , bac , acb , abc . The program hides the first card and displays the remaining two. If the ordered pair has not been used for a previous entry, then the program uses it and moves on to the next entry. Otherwise, it tries the next one of the six permutations, in order.

For example, the slot labeled 123 gives rise to the list 321, 312, 231, 213, 132, 123. We hide 3, display 2-1, and move on to the next slot. The next slot is 124 and gives rise to the list 421, 412, 241, 214, 142, 124. We try to hide 4 and display 2-1, but 2-1 taken, so we end up hiding 4 and showing 1-2. This continues on similarly. The neat thing about this method is that for this example, it never backtracks. It considers at most six choices for each entry, always succeeding by the sixth choice. This is fortunate but will it also succeed without backtracking for larger values of n ? Just as important, is there any simple computation underlying the strategy, which can be reproduced efficiently during a real time presentation by the magicians?

The theoreticians point out that even if the program is so lucky as never to have to backtrack, it still will need to examine $n! \cdot \binom{d}{n}$ tables in the worst case. For $d = 8$ and $n = 3$, this is a mere 336 tables. But for $d = 124$ and $n = 5$ it is already 1,350,900,144 tables. More importantly, how do we know that it never needs to backtrack? Maybe it just got lucky?

The hacker tries his program for $n = 4$ and $d = 27$, and the program returns a successful strategy in a modest few minutes. Unfortunately, it runs out of stack space when $n = 5$ and $d = 124$.

2.6. How Many Strategies for the Upper Bound? Is there a unique solution or strategy for handling the upper bound? It seems that there was flexibility in our choices when $n = 3$ and $d = 7$, but maybe when $d = 8$ all strategies are equal to the one we considered above? Silence reigns for a while, while more students work on their programs. A second student comes up with a new program that creates a strategy for $n = 3$ and $d = 8$ different from the first one shown above. The uniqueness conjecture is quickly shot down.

The second student looks at things from the other direction. There are $8 \cdot 7 = 56$ ordered pairs of eight numbers, and $\binom{8}{3} = 56$ unordered triples of eight numbers. Thus, in the maximal deck size case, a strategy is a bijection between these two sets. Thus, instead of listing the unordered triples, this second program lists the ordered pairs as the first row in the table. To do this, it generates all pairs $a - b$, with $a \leq b$, in lexicographic order, and then inserts the pair $b - a$ after each corresponding pair $a - b$. For each pair, the program

creates a table of unordered triples in lexicographic order, fills the first unused triple into the empty slot, and moves onto the next ordered pair. The result is shown below.

1 – 2	2 – 1	1 – 3	3 – 1	1 – 4	4 – 1	1 – 5
123	124	134	135	145	146	125
5 – 1	1 – 6	6 – 1	1 – 7	7 – 1	1 – 8	8 – 1
156	126	136	127	137	128	138
2 – 3	3 – 2	2 – 4	4 – 2	2 – 5	5 – 2	2 – 6
234	235	245	246	256	257	236
6 – 2	2 – 7	7 – 2	2 – 8	8 – 2	3 – 4	4 – 3
267	237	247	238	248	345	346
3 – 5	5 – 3	3 – 6	6 – 3	3 – 7	7 – 3	3 – 8
356	357	367	368	347	378	348
8 – 3	4 – 5	5 – 4	4 – 6	6 – 4	4 – 7	7 – 4
358	456	457	467	468	147	478
4 – 8	8 – 4	5 – 6	6 – 5	5 – 7	7 – 5	5 – 8
148	458	567	568	157	578	158
8 – 5	6 – 7	7 – 6	6 – 8	8 – 6	7 – 8	8 – 7
258	167	678	168	268	178	278

The first hacker stares for a while and announces that the new strategy is the same as the old if every ordered pair $x - y$ is replaced with $y - x$. Perhaps outside of that trivial difference, there are no other strategies?

The students are gaining interest. We still do not know whether these programs will generate strategies in general, nor do we know if either is guaranteed not to backtrack. At this point, we decide to take a look at the 52 card strategy. We finally explain the details of how we did it to the class.

3. THE 52-CARD STRATEGY ($d = 52$, $n = 5$)

The TA chooses which four cards to show, as well as their order. There are $\binom{5}{4}$ ways of choosing four cards from five, and $4!$ ways of ordering the cards. Hence she can send 120 different pieces of information. The hard part is that the professor cannot easily decode which of these 120 was sent. He can easily decode 24 pieces of information, but not so easily the extra factor of five.

One method that allows the professor to decode more than the obvious 24 is to notice, by the pigeonhole principle, that among the five cards chosen, there must be at least two of the same suit. The first card shown by the TA is one of these two cards, and the second is

the hidden card. Suppose we order the cards clockwise A, 2, 3, etc. through K in a circle, and define the distance between card x and card y , $distance(x,y)$ to be the length of the path clockwise from x to y . Then given any two cards, x and y , either $distance(x,y) \leq 6$ or $distance(y,x) \leq 6$. This is due once again to the pigeonhole principle: if they were each distance 7 or more from each other, then there would have to be at least 14 cards in the circle. The TA shows the card x such that $distance(x,y) \leq 6$. For example, given the 3 and the Jack, the TA shows the Jack, since $distance(J,3) = 5$.

The TA then chooses an ordering of the remaining three cards to encode a number from 1 to 6. The professor decodes this number and adds it to the first card in order to recover the identity of the missing card.

There are many ways to identify the $3!$ permutations of three cards with the numbers 1 through 6. We do it in a way that allows fast decoding, but any way is fine as long as a convention is agreed upon. Our order of the six permutations from 1 to 6 is: (123) (213) (231) (132) (312) (321). In order to decode a given permutation, the position of the lowest card tells us 1, 2 or 3. The order of the higher two cards, tells us whether or not to add 3 to this number: if they are in order, do not add 3; otherwise add 3. For the purposes of ordering we assume that if there is a tie then we break it by the bridge order of the suits which is Clubs, Diamonds, Hearts and Spades ($\clubsuit, \diamond, \heartsuit, \spadesuit$). This is also alphabetical order, in English. For example, the sequence $3\diamond, A\clubsuit, 2\heartsuit$ decodes to the number 5. This is because the smallest value is the A, which is in position two, and the remaining two cards $3\diamond, 2\heartsuit$ are not in order, so we add three to two giving the value 5.

3.1. A complete example. Suppose the TA is given the cards $4\diamond, 7\diamond, 9\spadesuit, J\clubsuit$, and $9\heartsuit$. Since there are two diamonds, the TA will hide either the $4\diamond$ or the $7\diamond$. Since $4\diamond$ is three below $7\diamond$, the TA shows the $4\diamond$, hides the $7\diamond$, and then arranges the remaining three cards to encode 3. This permutation is (231), so the TA ends up showing the four cards, in order, $4\diamond, 9\spadesuit, J\clubsuit$, and $9\heartsuit$.

From the other point of view, the four cards seen by the professor are in order: $4\diamond, 9\spadesuit, J\clubsuit, 9\heartsuit$. Because the first card is the $4\diamond$, the professor knows that the hidden card is between the $5\diamond$ and the $10\diamond$. The permutation of $9\spadesuit, J\clubsuit, 9\heartsuit$ is (231), the third

permutation on our list, so the professor adds three to the $4\heartsuit$. This gives the correct answer $7\heartsuit$ for the hidden card.

3.2. Generalizing the 52-card Strategy. Our strategy will not work for 53 cards. We use the choice of the first card shown to narrow the missing card down to one of six cards in a particular suit. We use the order of the last three cards to identify which one of the six cards. Altogether this allows us to identify exactly 48 cards. Adding in the four visible cards, this works for a deck of size at most 52 cards.

This can be generalized for any n . For example, when $n = 4$, we use only three suits. The deck would be divided into cards of suits A, B and C. Any set of four cards has to have at least two cards of the same suit. One of these cards is chosen as before, leaving two cards to be ordered. The number of orderings of two cards is two, so we can manage suits of length $2(2) + 1 = 5$. There are three such suits, making 15.

Our strategy works for any n , but the size of the deck for which it works is limited. The maximum size deck equals the number of suits, $n - 1$, times one more than twice the number of orderings of $n - 2$. This equals $(n - 1)(2(n - 2)! + 1)$.

3.3. What we know so far. Now we can update our progress:

Number of Cards Chosen	Lower Bound on Deck Size d	Upper Bound on Deck Size d
$n = 2$	3	3
$n = 3$	6 (8 by computer)	8
$n = 4$	15 (27 by computer)	27
$n = 5$	52	124
n	$(n - 1)(2(n - 2)! + 1)$	$n! + n - 1$

We have a simple computable strategy that improves on our original naïve lower bound strategy, but this still leaves quite a gap between the lower and upper bounds. We can actually reach the upper bound for the cases when n is 2, 3 or 4. However, the actual strategies for the latter two cases are computer-generated, and as far as we can tell, not easily memorized. Furthermore, we have no guarantee that these computer methods generalize to higher values of n .

The skeptical student in the class points out that there are redundancies in the 52 card model. For example, if the cards $A\spadesuit$, $5\spadesuit$, $6\heartsuit$, $Q\heartsuit$, and $3\clubsuit$ are drawn, then there are two choices for the set of four cards ($A\spadesuit$ $3\clubsuit$ $Q\heartsuit$ $6\heartsuit$) or ($6\heartsuit$ $5\spadesuit$ $A\spadesuit$ $3\clubsuit$). There are many other cases where there are multiple choices, and the TA just picks whichever one is easiest to calculate. It seems that we could manage our strategy more efficiently, avoiding these redundancies, and perhaps thereby handle a deck of greater than 52 cards.

The appendix describes a complicated method to move to a 53 card deck discovered by Weiping Shi [Sh]. It is an ad-hoc, ingenious, but complex method whose details are not as important as the fact that some method exists.

4. AN EXISTENCE PROOF: LOWER BOUND = UPPER BOUND

After learning the method for 53 cards, the class is discouraged and skeptical about finding a method for a 124 card deck. We reassure the class that although finding an explicit strategy for a 124 card deck may seem daunting, with the appropriate viewpoint and theorems, we can prove that such a strategy must exist.

There is a proof that a strategy exists for the maximal deck size, thereby showing once and for all, that the lower bound is equal to the upper bound. Consider the case where $n = 3$ and $d = 8$. In this case, the number of ordered pairs equals the number of sets of three cards from eight, both are equal to 56. A strategy can be thought of as a bijection between these two sets. Each unordered triple abc where $1 < a < b < c < 8$, can be assigned one of six ordered pairs, namely $a - b$, $b - a$, $a - c$, $c - a$, $b - c$, $c - b$. On the other hand, each ordered pair $x - y$ can be assigned to six unordered triples xyz for all $z \neq x$ and $z \neq y$. For example, $3 - 2$ can be assigned to 123, 234, 235, 236, 237, 238; and 234 can be assigned to $2 - 3$, $3 - 2$, $2 - 4$, $4 - 2$, $3 - 4$, and $4 - 3$. It helps to think about these bijections by abstracting the problem to a bipartite graph matching problem.

4.1. The Puzzle as a Graph. Let's consider the case of a deck of size d from which we choose n cards. Another way to think of a strategy is to model the problem with a graph. The nodes of the graph are the ordered subsets of size $n - 1$ and the unordered subsets of size n . The edges of the graph connect every unordered subset of size n with all of the

ordered subsets of size $n - 1$ that are contained in the unordered n -element set. Note that there are no edges between any two nodes both of which represent unordered n -element sets, and there are no edges between any two nodes both of which represent ordered $n - 1$ -element sets. All the edges connect ordered $n - 1$ -element sets to unordered n -element sets, and vice versa.

4.2. Bipartite Graphs, Perfect Matchings and Hall's Theorem. This kind of graph is called a *bipartite* graph, because you can partition the nodes into two parts, such that all the edges in the graph are between nodes in the two different parts. In our case, the sizes of the two partitions are equal. A *strategy*, from the graph point of view, is a collection of disjoint edges where every node is contained in exactly one edge. Such a collection of edges in a bipartite graph with equal size partitions is called a *perfect matching*. Finding a perfect matching in a bipartite graph is equivalent to the so-called *marriage problem*, where you are given a list of n men and n woman, and a list of pairs (man, woman) who are willing to get married, and you must find a subset of these pairs that allows each man and woman to be part of exactly one marriage.

There is a theorem about bipartite graphs with equal size partitions and perfect matchings called Hall's Theorem. The theorem states intuitively that unless something really obvious gets in your way, then you can always solve the marriage problem. What could get in your way? What if six men were willing to be paired with a total of only five women? Then you would be stuck for a perfect matching because there would not be enough women for that set of men. We state this more precisely.

Theorem 4.1 ([BM, 72–73]). *Suppose G is a bipartite graph, with a partition of equal size sets A and B . Then the following three facts are equivalent:*

- (1) *G has a perfect matching.*
- (2) *Every subset of k nodes in A connects to at least k nodes in B .*
- (3) *every subset of k nodes in B connects to at least k nodes in A .*

Technically, Hall's Distinct Representative theorem is more general than the one above. It works for any bipartite graph regardless of the degree and size of the two partitions,

and talks about matchings that are not necessarily perfect. It guarantees a matching that includes all the nodes on one side of a partition if and only if every subset of k nodes on that side connects to at most k nodes on the other side. In our situation, however, we only require the special case of Hall's theorem stated above.

4.3. What Does Our Graph Look Like? Our graph is a bipartite graph with equal size partitions. For general n and d , the number of nodes in each part of the bipartite graph is $\binom{d}{n}$ and $(n-1)! \cdot \binom{d}{n-1}$ respectively. The upper bound on the deck size d is given by $d = n! + n - 1$. The bipartite graph for such a case has $\binom{n!+n-1}{n}$ nodes in one part, and $(n-1)! \cdot \binom{n!+n-1}{n-1}$ nodes in the other. The reader can check that these are equal. The degree of each node in the first part is $\binom{n}{n-1} \cdot (n-1)! = n!$. The degree of each node in the second part is $d - (n-1) = n!$. The bipartite graph, however, is not symmetrical in the following sense. If you draw the graph with one partite set of vertices on the left, and the other set on the right, then the mirror image graph, with left and right sides reversed, is not isomorphic. For example, in the case $n = 3$, the ordered pair $1 - 2$ connects to the same six triples as $2 - 1$, but no two unordered triples connect to the same six ordered pairs. Fortunately, this asymmetry can be largely ignored.

4.4. Does our graph have a perfect matching? We prove by contradiction that our graph must have a perfect matching. The key point is that every node in our graph has the same degree; it is a *regular* graph. Assume there is a subset of k nodes on one side of the graph that connects to fewer than k nodes on the other side. The number of edges incident to the nodes in this subset is exactly $k \cdot n!$. If the number of distinct nodes incident to these edges on the opposite side of the graph is less than k , then by the extended pigeonhole principle one of the nodes on the opposite side of the graph must have strictly more than $\frac{k \cdot n!}{k} = n!$ edges incident to it. But we know that every node in the graph has degree $n!$, hence this is a contradiction. It is therefore impossible for a subset of nodes of size k to connect to fewer than k nodes on the other side. Therefore, Hall's theorem implies that there must be a perfect matching.

Our puzzle turns out to be a perfect matching problem in disguise. Because the graph of our puzzle is a regular bipartite graph it has a perfect matching. Therefore there exists

a strategy for the upper bound value of cards in a deck. Of course this implies that there are strategies for all smaller size decks as well, by just deleting the entries from the table using cards that were deleted from the deck.

The remaining questions now are:

- (1) Is there some memorizable or easily computable strategy?
- (2) How many different strategies are there for the maximal size deck?

5. A “MEMORIZABLE” OR “SIMPLY COMPUTED” STRATEGY

It is a struggle to find a strategy that is easy to compute or memorize. We can extend our 52 card strategy to 53, and ultimately 56 cards. For each extension, however, we must work very hard to glean enough extra information. See Section A for the 53 card strategy. The details of the 56 card strategy are left as a challenge to the reader.

The TA arrives one day and announces that she received an email from the friend who introduced her to this trick, and he has a strategy for $n = 5$ and $d = 124$ cards [K1]. It is even a memorizable strategy! Suppose you have a deck of 124 cards. Let’s think of them as numbers from 1 to 124. Given five of the numbers, the assistant adds the five numbers together, and divides by five. The remainder is a number between 0 and 4. If the remainder is r , then the assistant will *hide* the $(r + 1)^{\text{st}}$ number of the five, in ascending order. For example, if the TA sees the numbers 23, 27, 59, 87, and 93, these numbers have the sum 289, which is $4 \pmod{5}$. Thus, she removes the 5th number in the sequence, which is 93. She will show the remaining four numbers to the professor, in some order.

Using the remainder automatically reduces the number of possible answers we are looking for from 120 to $\frac{120}{5} = 24$. When the professor sees the numbers 23, 27, 59, and 87, he sums these to $1 \pmod{5}$. Thus, if the missing number occurs before 23, then the sum of the five numbers must have been $0 \pmod{5}$, and so the missing number must be $4 \pmod{5}$. Similarly, if the number is between 23 and 27, then the missing number must be $0 \pmod{5}$. If it is between 27 and 59, the missing number must be $1 \pmod{5}$. Between 59 and 87, it must be $2 \pmod{5}$, and above 87, the missing number must be $3 \pmod{5}$. We summarize this in the table below. The twenty-four X ’s signify the possible positions of the missing

number, the *’s signify the showing numbers, and the ·’s signify impossible choices for the missing number. The reader should check for herself that exactly 24 (and in general $(n - 1)!$) X ’s appear in this table.

1	2	3	4	5	6	7	8	9	10
·	·	·	X	·	·	·	·	X	·
11	12	13	14	15	16	17	18	19	20
·	·	·	X	·	·	·	·	X	·
21	22	23	24	25	26	27	28	29	30
·	·	*	·	X	·	*	·	·	·
31	32	33	34	35	36	37	38	39	40
X	·	·	·	·	X	·	·	·	·
41	42	43	44	45	46	47	48	49	50
X	·	·	·	·	X	·	·	·	·
51	52	53	54	55	56	57	58	59	60
X	·	·	·	·	X	·	·	*	·
61	62	63	64	65	66	67	68	69	70
·	X	·	·	·	·	X	·	·	·
71	72	73	74	75	76	77	78	79	80
·	X	·	·	·	·	X	·	·	·
81	82	83	84	85	86	87	88	89	90
·	X	·	·	·	·	*	X	·	·
91	92	93	94	95	96	97	98	99	100
·	·	X	·	·	·	·	X	·	·
101	102	103	104	105	106	107	108	109	110
·	·	X	·	·	·	·	X	·	·
111	112	113	114	115	116	117	118	119	120
·	·	X	·	·	·	·	X	·	·
121	122	123	124						
·	·	X	·						

At this point, a permutation will encode which of the possible 24 numbers is the hidden number. We will use the lexicographic order on the permutations for this encoding. That is, the permutation (1234) will encode the first X , the permutation (1243) will encode the second X , and the permutation (4321) will encode the last X . The TA wants to encode the 18th X , and the 18th permutation in lexicographic order is (3421). Thus, the TA shows the professors the numbers, in order, 59, 87, 23, and 7.

Now let’s work out an example from the other direction. Suppose that the professor sees the numbers 119, 32, 7, and 95. The sum of these four numbers is $3 \pmod{5}$. Thus, if the hidden number were to occur before 7, then the sum of all five of the numbers would have to be $0 \pmod{5}$, so he knows that the missing number would have to be $2 \pmod{5}$: it would have to be 2. If it were between 7 and 32, then it would have to be $3 \pmod{5}$: it could be 8, 13, 18, 23, or 28. If it were between 32 and 95, then it would have to be $4 \pmod{5}$. If it were between 95 and 119, it would have to be $0 \pmod{5}$, and finally if it were between 119

and 124, it would have to be $1 \pmod{5}$. Again, we summarize this in a table, as above.

1	2	3	4	5	6	7	8	9	10
.	X	*	X	.	.
11	12	13	14	15	16	17	18	19	20
.	.	X	X	.	.
21	22	23	24	25	26	27	28	29	30
.	.	X	X	.	.
31	32	33	34	35	36	37	38	39	40
.	*	.	X	X	.
41	42	43	44	45	46	47	48	49	50
.	.	.	X	X	.
51	52	53	54	55	56	57	58	59	60
.	.	.	X	X	.
61	62	63	64	65	66	67	68	69	70
.	.	.	X	X	.
71	72	73	74	75	76	77	78	79	80
.	.	.	X	X	.
81	82	83	84	85	86	87	88	89	90
.	.	.	X	X	.
91	92	93	94	95	96	97	98	99	100
.	.	.	X	*	X
101	102	103	104	105	106	107	108	109	110
.	.	.	.	X	X
111	112	113	114	115	116	117	118	119	120
.	.	.	.	X	.	.	.	*	.
121	122	123	124						
X	.	.	.						

Now, since the professor sees the numbers in the permutation (4213), this is the 21st permutation on the list, and so it corresponds to 105.

There are various tricks which allow this to be more easily computable. For example, suppose the numbers are in the order of the k^{th} permutation. Then the professor can compute the first X and add $(k - 1) \cdot 5$ that number. This will be close to correct. The professor must still make sure that he has a number with the correct remainder. To do this, he adds the number of $*$'s that are below the total sum. For example, if the professor sees 119, 32, 7, and 95, as above, then he knows the first possible X is 2. Then, because he has the 21st permutation on the list, he adds $20 \cdot 5 = 100$ to 2 to get 102. Finally he must correct the remainder: 7, 32 and 95 are below 102, so he adds 3 to get the final answer of 105 as the missing number.

6. OPEN QUESTIONS

Our class worked fulltime on this trick for half a week, but left many questions unanswered. The following list is left as a challenge for the reader and other classes.

- (1) Is there a way to determine whether a given search method will require backtracking?
- (2) How many different strategies are there for a given n and deck size d ?
- (3) How many of the strategies for some n and d are extendable to the maximal deck of size D ?
- (4) We can define equivalences on strategies. For example, any permutation of $n - 1$ will allow us to find a different, but somehow equivalent, strategy. Similarly, permutations of d will give us equivalent strategies, although they may not all be distinct. How many non-equivalent strategies are there?
- (5) How should we define a “simply computed” or “memorable” strategy?
- (6) How many of these strategies are there?

7. ACKNOWLEDGEMENTS

The events described in this paper are loosely based on our Discrete Mathematics class at ArsDigita University (<http://aduni.org>). We would like to thank all the people who contributed to this effort. Michael Kleber first introduced us to this trick. Colm Mulcahy generously shared his research about the origins of the trick. Students making guest appearances include Shyam Visweswaran, Chris Crick, and Heather Van Aelst, with a virtual appearance by Shyam’s wife. Weiping Shi provided the solution for the 53 card deck. Todd Sjoblum read and critiqued our drafts. An anonymous referee gave extensive helpful suggestions for revisions.

APPENDIX A. IMPROVEMENTS ON THE 52-CARD STRATEGY

Weiping Shi has found a clever strategy for a 53 card deck [Sh].

A.1. A Strategy for 53 Cards. Suppose we have a standard deck of cards, plus one additional card. Call the extra card the Jester (Je) of spades. From the professor’s viewpoint, he decodes the same way he did before, unless the ordered four cards he sees contain exactly two spades, one of which is the first card, that are separated by distance seven. From the TA’s viewpoint, she encodes in the same way except when the five cards she sees contain exactly two spades, one club, one diamond and one heart.

The details of the TA's encoding are described below, and the professor's decoding can be derived. The distribution of the five cards can be divided into one of the following cases:

- (1) There is at most one spade.

This implies (via the pigeon principle) that there must be at least two cards in another suit, because there are at least four cards in three suits. Use our regular method choosing two cards in the same suit and hiding one of them according to our usual procedure.

- (2) There are three or more spades, along with one or two cards in different suits.

Choose two spades at most of distance six apart, so that the third spade is not distance seven from the "higher" of the first two spades. This can always be done. We do this to avoid confusion with the next case. We use our regular method with the two chosen spades. We show the "lower" of the two spades. The remaining three cards indicate a number from one to six as before.

- (3) (a) There are exactly two spades, and there are at least two cards in another suit.

Use the regular method with the other suit and don't hide any spades.

- (b) There are exactly two spades, one heart, one diamond, and one club.

We can use the regular method except for the case when the two spades are distance seven apart, say A and 8. Then the TA shows both spades, one in the first position, and hides one of the other three cards. We will explain later how to choose which spade to show first and which card to hide. If the professor sees two spades of distance seven apart, then it has to be case 3b, and by looking at the other two suits among the four cards, he can immediately identify the missing card's suit.

The TA needs to choose an ordering of the last three cards, so that the identity of the missing card can be determined. The missing card and the appropriate ordering is chosen in this way. The cards A and 8 divide the range of values from A to Je into two parts, between A and 7, and between 7 and Je. Since we have three other cards, we know (again by pigeonhole) that two of them must lie within one of these parts. Assume that the two cards are in between A and 7, then the TA will show A as the first card. (If the two cards were in between

8 and Je, then we would show 8 as the first card, and proceed accordingly.) Either the lower card will have a distance of at most three from A, or the higher card will have a distance of at most three from 8. In the former case the TA hides the higher card, and in the latter case the TA hides the lower card. The other three cards are used to indicate the values +1, +2, +3 if we need to count up from A, or -1, -2, -3 if we need to count down from 8.

A.2. An Example for 53 Cards. For example, if the cards are $5\clubsuit$, $Q\heartsuit$, $A\spadesuit$, $8\spadesuit$, and $6\diamondsuit$, then the $5\clubsuit$ and $6\diamondsuit$ have values in between A and 8. The $6\diamondsuit$ is distance two from 8, so we hide the $6\diamondsuit$, and we show the $A\spadesuit$ as the first card. We order the remaining three cards to represent -2. (We will associate the permutations 123 132 213 231 312 321, with the values +1, -1, +2, -2, +3, -3, respectively.) Hence we encode the five cards $5\clubsuit$, $Q\heartsuit$, $A\spadesuit$, $8\spadesuit$, and $6\diamondsuit$ as $A\spadesuit Q\heartsuit 8\spadesuit 5\clubsuit$. To decode this, the professor sees that there are exactly two spades, where the $A\spadesuit$ is first and the $8\spadesuit$ is distance seven away. This implies case 3b, so the missing suit is diamonds and the missing rank is between A and 8. The permutation $Q\heartsuit 8\spadesuit 5\clubsuit$ represents 231, that in turn represents -2. Subtracting two from eight gives us six, so the missing card is the $6\diamondsuit$.

REFERENCES

- [BM] J.A. Bondy and U.S.R. Murty, *Graph theory with applications*, North Holland Publishing, 1982.
- [CL] G. Chartrand and L. Lesniak, *Graphs & Digraphs*, Third Ed. Chapman & Hall, New York, 1996.
- [G] M. Gardner, *Mathematics, Magic and Mystery*, Dover, 1956, p. 32.
- [KK] D. Kierstead and H. Kierstead, "Combinatorial Card Tricks," manuscript. See <http://dimacs.rutgers.edu/drei/1998/week1.html>.
- [K1] M. Kleber, personal communication.
- [K2] M. Kleber, *The Best Card Trick*, in preparation.
- [L] W. Lee, *Math Miracles*, Seeman Printery, Inc., Durham, NC, 1950, 49–51.
- [MSRI] Mathematical Sciences Research Institute newsletter, *The Emissary* <http://www.msri.org/ext/Emissary/EmissaryJan01.pdf>.
- [M1] C. Mulcahy, personal communication.
- [M2] C. Mulcahy, *Fitch Cheney's Five Card Trick*, to appear in Math Horizons, February 2003.
- [Sh] W. Shi, personal communication.

- [Si1] S. Simonson, *A Post-Baccalaureate Undergraduate Level Program in Computer Science*, On-Site Article, Communications of the ACM, Volume 45, No. 7, July 2002.
- [Si2] S. Simonson, "A Combinatorial Card Trick: Teaching discrete math to computer scientists should be fun." Lecture at the Eighth International Conference on Human Interface Technology, The University of Aizu, Aizu-Wakamatsu City, Japan, March 2002. For lecture notes, see <http://www.stonehill.edu/compsci/Japan.htm>

(Shai Simonson) DEPT. OF MATH AND COMPUTER SCIENCE, STONEHILL COLLEGE, N. EASTON, MA 02357

E-mail address: `shai@stonehill.edu`

(Tara Holm) DEPARTMENT OF MATHEMATICS #3840, UNIVERSITY OF CALIFORNIA, BERKELEY, CA 94720-3840

E-mail address: `tsh@math.berkeley.edu`