Fibonacci Solitaire and Its Use in the Classroom

Cliff F. Smith

June 16, 2010

This project was completed in partial fulfillment of the requirements for the Master's of Science in the Teaching of Mathematics at Portland State University.

Abstract

Fibonacci solitaire is a combinatorial algorithm devised by Gnedin & Kerov [8] to help examine the Young-Fibonacci graph, an example of a differential poset. The algorithm is explored and a method for finding the probability of a certain outcome is explained. The author describes several related activities that can be used in the classroom. Activities include exploration of permutations, functions, injective functions, function inverses, and combinatorial rules for counting. The activities are preceded by a discussion of how implementation of these activities in experimental classrooms led to modifications to the lessons.

Contents

1 The game

1.1 Directions for gameplay

Let n be a positive integer and assume n is even. To play the game, you will need a deck of cards numbered 1 to n. Shuffle the deck. Draw a card from the top of the deck and place it face up on the table in front of you. Draw the next card from the top of the deck and compare it to those on the table. The following rules dictate what to do with this card.

FS 1: If the drawn card is **smaller** in value than the smallest card on the table, place it on the table to the left of any cards on the table.

FS 2: If the drawn card is **larger** in value than the smallest card on the table, pair it with the smallest card on the table, removing them both from the table and placing this pair aside.

If at any point the table is empty, simply place the next card face up on the table and continue playing according to the previous rules. When all the cards have been played, the game is over. There may be cards remaining on the table. If no cards remain on the table, then all cards have been paired, and you win.

We will often have need to refer back to these original rules. When we do so, we will refer to them as **FS 1** and **FS 2**.

To clarify these instructions, we will consider an example game.

1.2 An example game

Here we will play an example game together using a six-card deck. It may be helpful to organize your deck as indicated and play through the example with your cards. We will refer to this game several times throughout our discussion. The deck is shuffled, and let us suppose that the cards in the deck have the following order: 4, 6, 5, 1, 2, 3. The first card, number 4, is turned over onto the table. The table appears as below:

On the table
4

The next card is 6, which is greater than 4. So, according to **FS 2**, we pair it with the 4 and set the pair aside. The table is now empty.

On the table	Removed from play
	(4,6)

The next card, 5, is placed on the table.

On the table	Removed from play
5	(4,6)

Next is card number 1. Since it is smaller than 5, we place it in the row to the left of the existing cards on the table in accordance with **FS 1**.

On the table	Removed from play
1 5	(4,6)

The 2 is drawn next, and since it is larger than 1, we pair it with the smallest card on the table, namely the 1. This pair is also set aside.

On the table	Removed from play
5	(1,2) (4,6)

Next we draw the 3, it is smaller than 5, so we place it on the table to the left of the 5 and our game is over.

On the table	Removed from play
3 5	(1,2) (4,6)

Since the 3 and 5 have not been paired, this is not a winning game.

1.3 Some questions

In this game, if you follow the rules, you don't have to make any strategic choices. There is no strategy involved, only an algorithm that guides you to the end of the 'game'. It is because of this algorithmic nature that we can consider the game as a function. From an initial starting point (a shuffled set of cards) we will end up at a uniquely determined end-of-game scenario, a pairing of cards. The resulting pairing is completely dependent upon the initial order of the cards in the deck. Some initial orders will lead to a win, others to a non-win.

It is easy for us to count the number of different orders of the cards in the deck, or permutations as we will call them from here on out. We will talk more about permutations later. After we permute the cards (yes, we can use it as a verb too) and before we flip over the first card, think about how many different values the first card can possibly have. Since there are 6 cards in our example, there are 6 possibilities for the first card. Then there are 5 possibilities for the second card because one possibility has already been taken by the first. This continues and, by the fundamental theorem of counting, we end up with $6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 6!$ possible permutations. In general, for an n-card deck, the number of permutations is given by $n!$

One obvious question is then: "How many permutations result in a winning game?" Once that number is found, we can answer the question: "What is the probability of winning?" Some students might ask: "Is the function defined by the given algorithm one-to-one? Is it onto?" Some of these answers are easily found for small deck sizes. For instance, if your deck has only 4 cards, then there are only 24 permutations. These can be checked to determine how many permutations result in a win. Larger deck sizes, however, are not as simple a matter, and a general solution is not immediately apparent. We will examine these and other questions later on. The driving focus will be to determine the probability of winning for an arbitrary, even deck.

2 Definitions and notation

Some terms that we will use have already been introduced, but have not yet been defined. Words like 'permutation' deserve a more formal definition. We will do that here as well as introduce some new vocabulary that we will use in further discussion.

Permutation: Informally, a permutation is a rearrangement of a set. More formally, a permutation of a set A is a bijective function $f : A \to A$. In our case, a shuffled deck of cards is a permutation of the set of cards. We will write permutations in brackets. Our permutation from the example game above is written $[4, 6, 5, 1, 2, 3]$, meaning that the first card drawn is 4, followed by 6, then 5, and so on. The number of permutations on a set of n objects is given by the *factorial* $n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 2 \cdot 1$

Pair: We will use this word only to describe cards that have been paired according to **FS 2** (or the act of pairing two cards by **FS 2**). We will denote such pairs (a, b) where a and b are the numbers of the cards, and where a is always smaller than b. For instance, suppose that, after playing the game, 4 was paired with 6 according to **FS 2**. When referring to this pair we would say "the pair $(4, 6)$" or simply "$(4, 6)$". We will never refer to the pair $(6, 4)$. The smaller card in a pair will sometimes be called the **bottom**, and the larger the **top**.

Singleton: We will use this word to describe cards that have not been paired as above. Such cards will be referred to by their card number by saying "the singleton 4", or simply "4". The smallest singleton will often be referred to as the **base**.

Pairing: We will call the result of any game–the set of pairs and singletons–a pairing. We will also have a standard notation for our pairings to help avoid confusion. The notation will also simplify the task of determining whether two pairings are the same. In this notation, pairs and singletons are written in numerical order, comparing the bottoms of pairs to the singletons. We will use our example game from above to help clarify. Remember that 3 and 5 remained as singletons and that $(4, 6)$ and $(1, 2)$ were the resulting pairs. This pairing is written "$(1, 2), 3, (4, 6), 5$". Note that the smallest of the bottoms and singletons, 1, is placed first in this notation, followed by the next largest, 3, and so on, until all pairs and singletons are written. We will call the result of a winning game a **perfect pairing**. A perfect pairing contains only pairs and no singletons.

Partial Pairing: It will become useful to discuss pairings that arise before the completion of a game, we will call these partial pairings. Such a pairing will contain a subset of the cards in the deck, and will be written according to our standard notation.

Double Factorial: It will be helpful at later stages to use the notation $n!!$ which we define as the product $n!! = n(n-2)(n-4)(n-6)\dots(n-2t)$ where t is the largest integer that satisfies $2t < n$. It is important to discern between $n!!$, as we have defined it, and $(n!)! = n!(n! - 1)(n! - 2)\dots(n! - (n! - 1))$.

2.1 Playing with new notation

Using the same permutation as in our initial example, $[4, 6, 5, 1, 2, 3]$, after our first card is drawn, we have the partial pairing 4. After drawing again we have $(4, 6)$. After the third card we get $(4, 6), 5$. Following this process and writing the pairs and singletons in order as described in the previous section we obtain the sequence of partial pairings:

$$4$$
$$(4, 6)$$
$$(4, 6), 5$$
$$1, (4, 6), 5$$
$$(1, 2), (4, 6), 5$$
$$(1, 2), 3, (4, 6), 5$$

Our final pairing consists of the pairs $(4, 6)$ and $(1, 2)$ as well as the singletons 3 and 5. The numbers 4 and 1 are bottoms with 6 and 2 their respective tops.

3 Playing backward

If you play the game a few times, it may become clear that a given pairing produced by the algorithm may come from more than one distinct permutation. As an example, consider a deck of four cards. If the initial permutation of these cards is $[4,2,3,1]$ in that order, then the result will be $1,(2,3),4$. Now, play the game again starting with the permutation $[2,3,4,1]$. You will again end at $1,(2,3),4$. Thus, the game, when viewed as a function from permutations to pairings, is not one-to-one. What if we wanted to be able to play backward? That is, starting from a pairing, how can we determine the permutation from which it arose, if any? Our understanding of the relationship between injective functions and inverse functions tells us that we cannot, from the pairing alone, determine the initial permutation. To do so, we would need some more information. What other information do we need?

It turns out that, to determine the initial permutation, it is sufficient to keep track of a bit of information concerning the bottoms in the final pairing. To do this, we will refer to the **depth** of a card. The depth is defined to be the number of cards written to its right, when the cards are written according to our standard notation. Recall that the result of the prior example game was $(1,2),3,(4,6),5$. In *this* pairing we say that the depth of 3 is 3, and that the depth of 4 is 2. It will be helpful to use a shorthand notation for this concept. Let $\delta(a)$ denote the depth of card a. So we will write $\delta(3)=3$ and $\delta(4)=2$.

We will need to talk about the depth at different steps of the game, so we need to modify the notation once more. Say we wanted to track the depth of card 4 at each step of the algorithm. We will use the notation $\delta_x(a)$ to denote the depth of card a when x cards have been drawn. Using our example again, recall the sequence of partial pairings:

Stage of game	Partial Pairing
1	4
2	(4,6)
3	(4,6), 5
4	1,(4,6),5
5	(1,2),(4,6),5
6 (final)	(1,2),3,(4,6),5

Referring to the above, we see that $\delta_1(4) = 0$, $\delta_2(4) = 1$, $\delta_3(4) = 2$ and so on. If a card has not been drawn at a given stage of the game, its depth at that stage is undefined. For instance, $\delta_1(1)$, $\delta_2(1)$, and $\delta_3(1)$ are all undefined, while $\delta_4(1) = 3$.

To be able to play backward, a record of the depth of cards at each stage would be useful to know. In fact, the depth of the bottoms of pairs are all we really need, and for these we need only to record their depth at the moment they were paired. Let's look at our example again. We need to know the depth of bottom b of pair (b, t) when b and t are first paired. That is, $\delta_x(b)$ where x is the step in the game when b is first paired with t. Careful thought indicates that the x^{th} card drawn must be t. Let's try to play our example game, backward. In order to do this we must collect the information discussed above. So, looking at our continued example we see,

Bottom	Step of game when bottom paired	Depth of bottom at that step
4	2	$\delta_2(4) = 1$
1	5	$\delta_5(1) = 4$

It will be helpful to define a new term to discuss the pairings along with this information about the depth of bottoms when paired. Let's call this information a **rigging**. Then by a **rigged pairing** we mean a pairing along with the rigging that will allow us to play backward. The algorithm for backward play can be stated as follows.

3.1 Reverse algorithm

Given a rigged pairing or rigged partial pairing, compare the current depth of all bottoms with their rigging.

Reverse 1: If no bottoms have a current depth matching their rigging, remove the smallest singleton.

Reverse 2: If any bottoms have a current depth matching their rigging, consider the smallest such bottom, b. If there are no singletons smaller than b, remove the top that is paired with b. Otherwise, remove the smallest singleton.

As cards are removed from the partial pairing according to these rules, place them face down in a pile, creating the original deck.

To see why this algorithm works, we return to our example pairing

$$(1, 2), 3, (4, 6), 5.$$

To eliminate the need to consult the previous table, we will write the rigging on the bottoms as a subscript. Thus,

$$(1_4, 2), 3, (4_1, 6), 5$$

indicates that the card 1 had $\delta(1) = 4$ when it was paired, and that 4 had $\delta(4) = 1$ when it was paired. These subscripts, the rigging, will remain with their respective card values throughout the reverse game.

3.2 A backward example

We will begin with the pairing resulting from our repeated example with the rigging we developed for it. Examining our rigged pairing,

$$(1_4, 2), 3, (4_1, 6), 5$$

we see that $\delta(1) = 5$ and $\delta(4) = 2$. So since neither of the depths of the bottoms match their rigging, we look at singletons. Since 3 is the smallest singleton, we remove it by **Reverse 1**. Our rigged partial pairing now looks like

$$(1_4, 2), (4_1, 6), 5.$$

Now we look at the depths again. Notice that $\delta(1) = 4$, which is the same as the rigging on 1. There are no singletons smaller than 1, that means that this is the step at which 1 was paired. We then remove the 2, leaving 1 as a singleton, by **Reverse 2**.

$$1, (4_1, 6), 5$$

At this point, none of the depths of the bottoms matches their depth when paired. So we remove 1 in accordance with **Reverse 1**. We are left with

$$(4_1, 6), 5.$$

We see that the depth of 4 still does not match its rigging. So we remove 5, by **Reverse 1**, leaving only the pair

$$(4_1, 6).$$

Now finally, $\delta(4) = 1$ so this is the step at which 4 was paired. We remove its top, 6 because that's what **Reverse 2** says to do. Then the only card left is

$$4.$$

It is the smallest singleton, so we remove it by **Reverse 1**.

Thus we have arrived at our original permutation, and if we play the game, starting with this permutation, because our game is a function, we will once again obtain our pairing, $(1, 2), 3, (4, 6), 5$. We see then that using the rigging, we can play the reverse game and obtain our original permutation again. Notice that at each step of the reverse game, we obtained a rigged partial pairing that corresponds to a partial pairing in the sequence of partial pairings developed previously, and that we obtained them in the reverse order.

Stage of Reverse Game	Rigged Partial Pairing	Deck
1	$(1_4, 2), 3, (4_1, 6), 5$	
2	$(1_4, 2), (4_1, 6), 5$	[3]
3	$1, (4_1, 6), 5$	[2, 3]
4	$(4_1, 6), 5$	[1, 2, 3]
5	$(4_1, 6)$	[5, 1, 2, 3]
6	4	[6, 5, 1, 2, 3]
7		[4, 6, 5, 1, 2, 3]

Next, we will prove that these reverse rules work for any rigged pairing.

3.3 Playing backward is an inverse function

Consider an arbitrary game with a deck of n cards at stage p, meaning that p cards have been drawn and processed according to the original rules. So the number of cards on the table is p or fewer, and there may be a smallest unpaired card, the base, call it b_p. When we draw the $p+1^{st}$ card, either it will be larger or smaller than our base b_p.

3.4 Case 1: New base

If the $p+1^{st}$ card is smaller than b_p, or if b_p does not exist, then the $p+1^{st}$ card becomes the new base, called b_{p+1}, and we move to the next draw. Consider what would happen at this point if we were following our reverse algorithm. The base b_{p+1} is the smallest singleton and there may be other singletons and pairs. Some of the pairs may have riggings matching their current depths. However, those pairs must have bottoms larger than b_{p+1} because when b_{p+1} was placed according to our standard notation, the depth of any pair with a smaller base would have increased. So, according to **Reverse 2**, the next card we must pick up is the smallest singleton, b_{p+1}, as desired.

3.5 Case 2: Pairing

If the $p+1^{st}$ card is larger than b_p, it gets paired with the base b_p, which becomes a bottom, and the new card is a top which we will denote t_{p+1}. We associate with b_p a rigging, namely $\delta_{p+1}(b_p)$. When we consider what would happen at that particular step of the reverse game, we see that b_p currently has a depth matching its rigging. All bottoms smaller than b_p do not have riggings matching their current depth because of the addition of t_{p+1}. Further, there are no singletons smaller than b_p because it was the smallest singleton at the moment it was paired. Therefore, by **Reverse 2** we must pick up the top of this pair, t_{p+1}.

Thus, whenever we move from the p^{th} to the $p+1^{st}$ step of the game, our reverse algorithm will guide us back from the $p+1^{st}$ to the p^{th} step. We conclude that the reverse algorithm is therefore the inverse of Fibonacci Solitaire.

4 Appropriate bounds on the rigging

What values are allowable on the rigging for our purposes? We can obviously state that any rigging must be comprised of non-negative integer values because depth is always given as a non-negative integer. It is also fairly clear that, since we are counting the corresponding top card when determining the depth of a given bottom, the rigging must also be greater than zero. Thus, values for a rigging must come from the set of counting numbers. What is an appropriate upper bound for the rigging, r_b, on a bottom, b, in any given pairing?

4.1 Why $r_b \leq \delta(b)$

Suppose a pairing, say for example $(1, 2), 3, (4, 6), 5$ had the following rigging $(1_2, 2), 3, (4_5, 6), 5$. Let's follow our reverse algorithm and see where this leads. None of the riggings match the bottoms' current depths, so we must remove the 3 by **Reverse 1**. We are left with $(1_2, 2), (4_5, 6), 5$ and still no riggings match, so we remove the 5. At this point, $(1_2, 2), (4_5, 6)$, we have reached an impasse. None of the riggings match, yet there are also no singletons that we can remove. Clearly, this is not a valid rigging for the given pairing, but why?

Considering the same pairing, but without rigging, $(1, 2), 3, (4, 6), 5$, we can clearly see $\delta(4) = 2$ and $\delta(1) = 5$. If we now start removing cards, it is only possible that the depths of 4 and 1 will decrease, they cannot increase. But in our example above, the rigging on 4 was 5. The depth of 4 will never exceed 2 since we are trying to remove cards. We can see that an appropriate upper bound for the riggings is the depth of the bottoms at the final pairing. For example, $(1_1, 2), (4_2, 6), 5$ is a valid rigging, while $(1_0, 2), (4_1, 6), 5$ and $(1_6, 2), (4_2, 6), 5$ are not. For any pairing or partial pairing, and for any bottom b, the rigging r_b on b must lie in $1 \leq r_b \leq \delta(b)$.

4.2 Are all such riggings valid?

The question then becomes: If r_b lies in that interval, will the rigged pairing yield a permutation through the reverse algorithm? Consider a rigged pairing where the values of the rigging on each bottom lie within the bounds as described above. It is possible that one or more of the bottoms initially has a rigging matching its current depth. If this is the case, such a pair will soon be split, and the bottom will remain as a single card. Since some bottoms have riggings less than their current depths, by removal of singles and other tops, the depths of these bottoms will decrease by increments of one. So, at some point the rigging of each bottom will match its depth. That pair will soon be split. If any pair has rigging 1, it will wait until there are no other pairs or singletons written to its right. At that point, the rigging will match the depth. It is because we only remove one card at a time, that the depth of any bottom will decrease only in increments of one. This guarantees that, at some point, the depth of a card will match its rigging.

5 Probability of winning

In this chapter, we will prove our main result, which computes the probability of winning Fibonacci solitaire.

Theorem 1 *Let n be even, and let $\sigma(n)$ be a permutation of the set $\{1,\ 2,\ 3\ \ldots,\ n\}$. The probability that $\sigma(n)$, under the Fibonacci solitaire algorithm, yields a perfect pairing is:*

$$Pr_{\text{win}}(n) = \frac{n!}{2^n \cdot \frac{n}{2}! \cdot \frac{n}{2}!}$$

Before we embark on the proof of the above result, let us recall what we mean by the probability of winning. Elementary probability theory tells us, given a deck of n cards, that the probability of winning corresponds to the number given by,

$$Pr_{\text{win}}(n) = \frac{\#\text{ of permutations resulting in a win}}{\#\text{ of permutations}}.$$

This quantity will lie between 0 and 1. The closer it is to 1, the more likely we are to win Fibonacci solitaire.

Now, if we consider the bijective nature of the Fibonacci solitaire algorithm with rigging, we conclude that the number of rigged perfect pairings is exactly the number of winning permutations. The probability in question is now,

$$Pr_{win}(n) = \frac{\#\text{ of rigged perfect pairings}}{\#\text{ of permutations}}.$$

In the sections ahead, we will use familiar counting rules to determine the number of riggings on a given pairing. We can also find the number of possible perfect pairings (wins). Using these two numbers, we will find the number of rigged perfect pairings. Our basic form for the probability of winning becomes,

$$Pr_{\text{win}}(n) = \frac{\#\text{ of perfect pairings} \cdot \#\text{ of riggings on a perfect pairing}}{\#\text{ of permutations}}. \quad (1)$$

There are many characterizations of this probability, depending on what logical thought processes were used in counting the perfect pairings. Below we will outline one argument for the number of riggings on a perfect pairing. We will then discuss several different logical pathways for counting the number of perfect pairings. In a subsequent section, we will prove that the characterizations of probability given by these counting arguments are equivalent to the probability given in Theorem 1.

5.1 The number of riggings on a perfect pairing

Lemma 1 *Given any perfect pairing of n cards, there are exactly*

$$(n-1)!!$$

permutations of n cards that result in the given perfect pairing.

Given a deck of n cards, a winning game will result in $n/2$ pairs. To determine the number of riggings on a set of $n/2$ pairs, we must first decide the depths of the n bottoms. Using $(a, *), (b, *), (c, *), (d, *)$ as a stand-in for a perfect pairing of an 8-card deck, where a, b, c, and d are the bottoms of the pairs, and the tops are designated $*$, we see that $\delta(a) = 7$, $\delta(b) = 5$, $\delta(c) = 3$, and $\delta(d) = 1$. We can choose a value of the rigging on a from the set $\{1, 2, 3, 4, 5, 6, 7\}$, allowing seven possible choices. For the rigging on b we must choose from the set $\{1, 2, 3, 4, 5\}$, thus we have five possible values we can choose. Continuing in this manner, we find that the total number of riggings possible on a perfect pairing of 8 cards is $7 \cdot 5 \cdot 3 \cdot 1 = 7!!$.

Generalizing the previous argument, consider a perfect pairing of a deck of n cards. Let b_a be the base of the a^{th} pair when the perfect pairing is written in our standard notation. Examining the perfect pairing,

$$(b_1, *), (b_2, *), (b_3, *), \ldots, (b_{n/2}, *)$$

we can see that the possible choices for a rigging on the bottom b_1 must come from the set $\{1, 2, 3, \ldots, (n-1)\}$. Similarly we can choose a rigging on the bottom b_2 from the set $\{1, 2, \ldots, (n-3)\}$ giving $(n-3)$ choices. Continuing on this line of thought, we see that for each perfect pairing, there are

$$(n-1)(n-3)(n-5)\ldots(n-(n-1)) = (n-1)!!$$

possible riggings for a winning game with a deck of n cards.

5.2 The number of perfect pairings

Now we must determine how many perfect pairings exist. There are several counting arguments that can be made that result in equivalent characterizations of the number of perfect pairings. Below we will describe three of these arguments and share the characterization of probability that arises from each.

5.2.1 The first method for counting perfect pairings

Lemma 2 *There are exactly*

$$\frac{n!}{\frac{n}{2}! \cdot 2^{\frac{n}{2}}}$$

distinct perfect pairings of n cards.

We can describe a set of $n/2$ pairs by pairing the numbers as they appear in any permutation. For example, when $n = 8$, the permutation $[4, 6, 7, 2, 8, 1, 3, 5]$ would correspond to the set of pairs $(4, 6), (7, 2), (8, 1), (3, 5)$. However, $n!$ over-counts our set of perfect pairings because it allows

$$(4, 6), (7, 2), (8, 1), (3, 5)$$

and

$$(8, 1), (7, 2), (3, 5), (4, 6)$$

to be counted as distinct pairings whereas, in our context, we view them as equivalent. For each of the $n/2$ pairs in a perfect pairing on n cards, there are $(n/2)!$ equivalent perfect pairings that result from permutations of the pairs. Correcting for this over-counting requires a division,

$$\frac{n!}{(n/2)!}.$$

We are not quite done, however, because we have only corrected for permutations of the pairs. The counting argument thus far, still counts the perfect pairings

$$(4,6),(7,2),(8,1),(3,5),$$

and

$$(6,4),(2,7),(1,8),(5,3)$$

as distinct. For each perfect pairing, the pairs themselves can be written with the smaller or larger card first. For each pair, this results in two options, small card first, or large card first. Since there are $n/2$ pairs in a perfect pairing of n cards, there are $2^{n/2}$ equivalent perfect pairings resulting from reordering the large and small card within each pair. To remove the over-counting of these instances, we must divide our previous count by $2^{n/2}$. Then we can express the number of possible perfect pairings as,

$$\frac{n!}{\frac{n}{2}! \cdot 2^{\frac{n}{2}}}. \tag{2}$$

5.2.2 A second method for counting perfect pairings

Consider a perfect pairing written according to our notation. The pair with the smallest bottom is written first, followed by the pair with the next smallest, and so on. For an even deck, since all cards have been paired, the smallest card must obviously be in a pair, thus this pair has the smallest bottom. Let b_1 represent the smallest of the n cards in our deck. Then we have $(n-1)$ choices for the top of this pair. Let b_2 represent the smallest unused card. Obviously, it too must be in a pair. Since b_2 is the smallest unused card, it is smaller than whatever card it will be paired with. Therefore, b_2 will be a bottom. We have $(n-3)$ choices for the top of this pair. Continuing in this fashion, we argue that each bottom b_a where $a \in \{1, 2, \ldots, (n/2)\}$ is uniquely determined from prior choices for tops of pairs. Therefore, the only things that need be counted are the choices available for those tops. We find that the number of ways to construct our perfect pairing is

$$(n-1)!! \tag{3}$$

.

5.2.3 A third method for counting perfect pairings

Another counting method involves the use of the **combination** function. For a deck of n cards, a perfect pairing consists of $n/2$ pairs. We have $\binom{n}{2}$ ways to choose one of these pairs, meaning that we must choose two numbers from our total set of n cards. Whichever two cards we choose, one will be smaller than the other, and it will become the base of this pair. Since we have already used 2 of the n cards available, there are $n-2$ cards remaining from which we can choose the next pair, the number of ways to do so is given by $\binom{n-2}{2}$. There are then $\binom{n-4}{2}$ ways to choose a third pair, and so on. We end at $\binom{n-(n-2)}{2}$. This method over-counts the number of perfect pairings in the same way that our first method did. For instance, if we were to choose $(1, 2)$ as our first pair and $(3, 4)$ as our second, nothing would have prevented us from choosing these same pairs in the reverse order, $(3, 4)$ then $(1, 2)$. Since we consider different permutations of the same pairs as equivalent we need to correct for permutations of the pairs. We must then take our product of combinations divided by $(n/2)!$ as the number of possible perfect pairings. This results in the following number of perfect pairings,

$$\frac{\binom{n}{2} \cdot \binom{n-2}{2} \cdot \binom{n-4}{2} \cdot \ldots \cdot \binom{n-(n-2)}{2}}{\frac{n}{2}!}. \tag{4}$$

5.3 Alternate characterization of $n!!$

In order to simplify computation, it is desirable to find an expression of $n!!$ that allows input into common calculators and computer algebra systems. We can easily write the double factorial using the familiar indexed product notation. For even n, we can say

$$n!! = \prod_{i=1}^{\frac{n}{2}} 2i.$$

We see that the factors in the product are all even because each is a multiple of 2. If we remove a factor of 2 from each factor in the product we obtain,

$$n!! = \prod_{i=1}^{\frac{n}{2}} 2i = 2^{\frac{n}{2}} \prod_{i=1}^{\frac{n}{2}} i = 2^{\frac{n}{2}} \cdot \left(\frac{n}{2}\right)!.$$

5.4 Determination of probability

We are now equipped to prove Theorem 1.

Proof of Theorem 1. The formula for the probability of winning, (1), together with Lemma 1 and Lemma 2 allow an expression from which we derive the desired result.

$$
\begin{aligned}
Pr_{\text{win}}(n) &= \frac{\frac{n!}{\frac{n}{2}! \cdot 2^{\frac{n}{2}}} \cdot (n-1)!!}{n!} \\
&= \frac{(n-1)!!}{\frac{n}{2}! \cdot 2^{\frac{n}{2}}} \\
&= \frac{(n-1)!! \cdot n!!}{\frac{n}{2}! \cdot 2^{\frac{n}{2}} \cdot n!!} \\
&= \frac{n!}{\frac{n}{2}! \cdot \frac{n}{2}! \cdot 2^{\frac{n}{2}} \cdot 2^{\frac{n}{2}}} \\
&= \frac{n!}{2^n \cdot \frac{n}{2}! \cdot \frac{n}{2}!}
\end{aligned}
$$

Remark. Using the counting arguments, from Sections 5.2.2 and 5.2.3, for the number of perfect pairings we can derive the same result.

Taking now the counting argument (3), from Section 5.2.2, we show

$$
\begin{aligned}
Pr_{\text{win}}(n) &= \frac{(n-1)!! \cdot (n-1)!!}{n!} \\
&= \frac{n!! \cdot n!! \cdot (n-1)!! \cdot (n-1)!!}{n!! \cdot n!! \cdot n!} \\
&= \frac{n! \cdot n!}{2^{\frac{n}{2}} \cdot 2^{\frac{n}{2}} \cdot \frac{n}{2}! \cdot \frac{n}{2}! \cdot n!} \\
&= \frac{n!}{2^n \cdot \frac{n}{2}! \cdot \frac{n}{2}!}.
\end{aligned}
$$

Finally, examining the number of perfect pairings, (4), given in Section 5.2.3 we see,

$$
\begin{aligned}
Pr_{\text{win}}(n) &= \frac{\binom{n}{2}\cdots\binom{n-(n-2)}{2}\cdot(n-1)!!}{\frac{n}{2}!\cdot n!} \\
&= \frac{\frac{n!}{2!\cdot(n-2)!}\cdots\frac{(n-(n-2))!}{2!\cdot(n-(n-2)-2)!}\cdot(n-1)!!}{\frac{n}{2}!\cdot n!} \\
&= \frac{n!\cdot(n-1)!!}{2^{\frac{n}{2}}\cdot\frac{n}{2}!\cdot n!} \\
&= \frac{(n-1)!!\cdot n!!}{2^{\frac{n}{2}}\cdot\frac{n}{2}!\cdot n!!} \\
&= \frac{n!}{2^{\frac{n}{2}}\cdot 2^{\frac{n}{2}}\cdot\frac{n}{2}!\cdot\frac{n}{2}!} \\
&= \frac{n!}{2^{n}\cdot\frac{n}{2}!\cdot\frac{n}{2}!}.
\end{aligned}
$$

6 Connecting the algorithm to Fibonacci

We can think about the sequence of partial pairings we outlined in our main example as binary strings where each singleton corresponds to a 1 in our string and each pair corresponds to a 2. It is important to discern between these binary strings, and the more familiar numbers from the decimal system. The symbol 11 in this section is read "one one", not "eleven".

Partial Pairing	Binary string
4	1
(4,6)	2
(4,6), 5	21
1, (4,6), 5	121
(1,2), (4,6), 5	221
(1,2), 3, (4,6), 5	2121

Each of the binary strings represents the order of pairs and singletons in the pairing. Notice that the sum of the numbers in the string equals the number of cards that have been drawn. So, while the length of our pairings is constantly increasing, it is not the length of the binary string, but the sum of its constituent digits that is constantly increasing.

This relationship can be discussed as a function *bin* from the set of finite pairings to the set of binary strings. The function *bin* is not injective as we can illustrate by example. The pairings

$$(1, 2), 3, (4, 6), 5$$

and

$$(1, 6), 2, (3, 5), 4$$

are clearly distinct, but their binary strings

$$bin\,[(1, 2), 3, (4, 6), 5] = 2121$$

and

$$bin\,[(1, 6), 2, (3, 5), 4] = 2121.$$

are equivalent. Thus, for each binary string there may be more than one corresponding pairing.

Interestingly, many of the rules of our Fibonacci solitaire game still apply. The possible sequences of our binary string are governed by the algorithm. For instance, the sequence below is impossible when we consider binary strings as representative of pairings.

$$1$$
$$2$$
$$21$$
$$121$$
$$1211$$
$$2121$$

Let us examine why this sequence does not correlate to a sequence of pairings that would arise from our algorithm. The only difference between this sequence and the example sequence at the beginning of this section is the fifth term. When we ask how we might move from 121 to 1211 according to our algorithm, we see that we would draw a card that is larger than the bottom of the pair. However, if this card is larger than the bottom of the pair, then it is also larger than the base which corresponds to the leftmost 1. According to **FS 2**, it would have been paired with the base, resulting in the binary string 221, rather than 1211.

Below is another sequence that is impossible, though for a slightly different reason.

1

2

21

121

122

2121

The problem arises in the same location as in the previous impossible sequence. We now see 121 followed by 122. In the context of the Fibonacci solitaire algorithm, this means that the card was paired with the singleton that was to the right of the pair. Because that singleton was to the right of the base, the leftmost 1, we know it must have been larger than the base. Therefore, the card drawn was bigger than the smallest card. Again, by **FS 2** the card would have been paired with the smallest card on the table which is the leftmost 1. In this case then, 121 should be followed by 221 instead of 122.

Given a partial pairing, we can devise a set of conditions that will allow us to determine the possible forms of the subsequent partial pairings. Stated differently, given a binary string, we will be able to determine the possible strings that can come next in the context of Fibonacci solitaire.

6.1 Binary algorithm

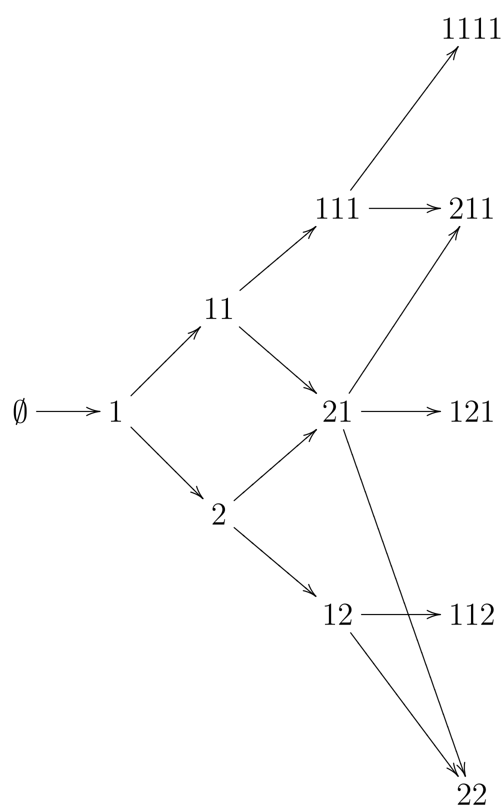
Binary 1: A 1 can be inserted anywhere to the left of all existing 1s.

Binary 2: The leftmost 1 can be changed to a 2.

We can see how closely these rules mirror the Fibonacci solitaire algorithm. **Binary 1** comes directly from the understanding that whenever a card is added as a singleton, it must be smaller than all other singletons. **Binary 2** is clearly related to **FS 2** in that, if a pair is created, it is the smallest singleton that gets paired with the most recently drawn card.

6.2 The Young-Fibonacci graph

Using these rules we can create a graph that indicates the forms of the partial pairings in all possible games. Starting with $\emptyset$, which we will use to represent our game before the first card has been drawn, the only possible next step is that we obtain the binary string 1. This corresponds to drawing the first card and simply placing it on the table. Next, we have two options, we can insert a 1 to the left of the existing 1, or we can replace the 1 with a 2. These two options result in the binary strings 11, and 2. These binary strings correspond, respectively, to drawing a second card smaller than the initial card, and to drawing a larger card, thus creating a pair. Continuing with this line of reasoning we construct the directed graph below. Each of the vertices is a binary string that corresponds to a number of pairings. Perfect pairings are indicated by binary strings consisting only of 2s. Each edge indicates a possible move between strings according to the binary rules above which correspond to **FS 1** and **FS 2**. For example, there are edges between 1 and 11 and between 1 and 2 because those are possible subsequent strings according to the binary algorithm.



The graph above is only a very small subgraph of the Young-Fibonacci graph. The Young-Fibonacci graph is infinite, having as its vertices *every* binary string. Moving left to right, look carefully at the number of vertices at each step of the graph. The number of vertices at each step corresponds to the Fibonacci sequence, {1, 1, 2, 3, 5, 8, 13,...}.

This graph shows the connection of the Fibonacci solitaire algorithm to the field of Graph Theory. Exploring the properties of this graph would be beyond the scope of this project, and beyond the comprehension of most high school students. Instead, we now turn to a discussion of how we might use the Fibonacci solitaire algorithm in a classroom.

7 Introduction to classroom activities

Fibonacci solitaire is a card game that begins with a shuffled deck and results in a pairing of the cards. The game offers opportunities for students to explore some abstract mathematical concepts while maintaining a concrete context. For example, the algorithmic card game can be viewed as a function. Students are encouraged to think about functional processes by playing the game.

Each of the lessons includes implementation suggestions for the instructor, student activity sheets, and a complete set of example solutions.

The first three lessons are preliminary to the game, which will not be introduced until Lesson 4. Lessons 1-3 introduce permutations. Students will shuffle a deck of cards to develop understanding of the concept and will practice with some notation. Lesson 4 introduces the Fibonacci solitaire algorithm. Students will learn to play the game and learn standard notation for writing results. Lesson 5 has students play several games and compute experimental probabilities of winning for two sample sizes. After developing a theoretic probability for small deck sizes, students will compare experimental and theoretic probabilities and make a conjecture about how sample size affects experimental probability. Lesson 6 revisits the algorithm with a focus on determining whether the function is injective. Students will have an opportunity to understand the connection between the injective property and invertible functions. Lesson 7 introduces some new notation and an algorithm that reverses the Fibonacci solitaire process. Students will understand that by the addition of some information, some functions can be made invertible. Lesson 8 concludes with a determination of the theoretic probability for playing with 8 cards.

7.1 Lesson implementation

These activities were designed, expecting group work and student effort to be the classroom norm. Students should be capable of working together and working to make sense of mathematics. Many of the lessons have natural breaking points where, depending on the need, an activity can be stopped for the day or paused to allow the class to reconvene and discuss what they are learning. The instructor may wish to distribute only part of a worksheet, to help with determining of a proper time to bring the class back as a whole. The following time line is generally applicable to all the lessons herein. The cyclic nature of the time line offers a flexibility that allows the instructor to break into group work and regroup as a class as many times as is necessary. Instructors whose classrooms do not have group work as a norm should clearly indicate expected behavior for working in groups and for the process of reconvening as a class. Indicate to your students that they will be expected to shift from their groups, to the teacher, and back when instructed. This method of classroom management allows greater opportunity for students and teachers to work together directly and removes much of the fear that students may have of looking foolish in front of the entire class. The cycle of instruction below always starts with preparation, on the teacher's part, before student contact. When students arrive, the cycle, “Direct, Monitor, Discuss” can be repeated as many times as necessary and allows flexibility for multiple class time periods.

Do This	Time	Description
Prepare	20 min	Read and attempt all activity problems. Examine the solutions sheet for example answers. Prepare for any questions you think your students may have about the activity. Gather required materials. Print work sheets for students or prepare overhead copy.
Direct	5 min	Give the students instructions about how you want them to complete the activity. If you want the students to work in groups, create those groups now. If you wish, you may divide the activity into a few parts to develop a natural return to full classroom discussion. Give them either the first part, “What is a permutation?”, or the whole activity.
Monitor	10-30 min	Move about the classroom. Listen to student discussion. Encourage students to read directions carefully and try to work out the concepts together.

Do This	Time	Description
Discuss	10 min	Reconvene as a class. Discuss any major misunderstandings you observed while moving about the room. Answer any questions students may have. Share an example. Ensure everybody understands what they should be doing. If you decided to break the activity into parts, distribute the second part if students are now ready.
Direct	1 min	Guide student attention from whole class discussion, to group discussion as they continue with the activity.
Monitor	10-30 min	After a clarifying classroom discussion, students may feel more confident that they can accomplish the tasks. Move about the classroom again, ensuring students are engaged in mathematical activity. If many students are encountering the same or similar problems, you may wish to reconvene the class one more time. Otherwise, let the students work to complete the activity.
Discuss	5-10 min	Always end the class by reconvening and clarifying any last minute misunderstandings. It is important to review what happened during the day's lesson to remind students what to pay attention to.

8 Lesson 1: Permutations

Lessons 1-3 are preliminary to the introduction of Fibonacci solitaire, which occurs in Lesson 4.

This lesson uses card shuffling as a method to introduce permutations. The use of cards to help explain the concept is quite helpful. A student might draw an association between shuffling a deck of cards, and permuting a set. Such an association could prove helpful in retaining the concept long after the class in which the concept was learned. The word permutation is commonly used in the English language as well, so helping students understand the term mathematically will hopefully allow them to extend their understanding to other areas of their life. Defining the term now will allow better understanding of the Fibonacci solitaire algorithm as a function. Understanding a function's domain is an integral part of knowing how the function behaves generally. Further, the Fibonacci solitaire algorithm as described uses two different notations for permutation. The *sequence* notation defined in the following lesson is used to describe elements of the domain, while *cycle* notation is how the algorithm defines elements of its range. Helping students see this now allows the use of the term permutation in later lessons, allowing simpler descriptions and greater likelihood of students understanding instructions.

Notes to the instructor

Materials *Cards:* Students will need a deck of numbered cards. A deck of size 10 is definitely sufficient. Many of the subsequent lessons require only six cards. A deck of standard playing cards, partitioned into suits, with King, Queen, and Jack removed will give 4 sets of 10 cards when the Ace is considered as a 1. If playing cards, or simply numbered cards are unavailable, you can have students make their own cards using index cards or slips of paper.

Lesson worksheets: The activity below can be photocopied and should be distributed to students. An overhead can be made from the activity as well, if so desired, then students will work with their own paper to answer the questions projected.

Goals Students will be able to describe what a permutation is.

Students will be able to write a given permutation according to two different notations.

Students will conjecture about the number of permutations on a set of size n.

Prerequisite student knowledge Students should be at a high school mathematics level.

Possible student misconceptions While sequence notation is very simple, Cycle notation can be deceptively difficult to grasp. Students may view the parentheses as meaning that those cards "go together" and may then simply place the cards as if in sequence notation. Those students who understand that those cards in the parentheses switch places may not be able to determine where the last card in the group goes. For instance, when presented with the permutation (1,4,2,3) 5 6, a student might not know what to do with the 3.

Activity 1 Permutations

May 2010 Cliff Smith

Name: ____________________________________

What is a permutation?

Informally, a *permutation* of a set is a re-ordering of that set. Shuffling a deck of cards offers a good example of a permutation. There are many different ways to talk about and write permutations. We will explore a few of them here. Start with the deck of cards your teacher gave you. Put the deck in ascending numeric order. When you draw from the top of the deck, the first card you draw should be 1, followed by 2, and so on. This permutation is going to be our starting point. We will write permutations in relation to this order as well as in relation to each other.

1. Shuffle the deck a few times, then draw the cards one at a time from the top of the deck. Use the table below to keep track of the order in which you draw the cards.

1st card	2nd card	3rd card	4th card	5th card	6th card

2. The order in which you wrote the cards in the previous problem is one way to write a permutation. For clarity, we will call writing permutations in this way ***Sequence Notation***. Simply write the numbers in the order in which they appear. Do it twice more here on your own. Shuffle the deck, then draw the cards once at a time and write them in the order they appear. Put all the numbers into one set of [brackets]

3. Did you get the same permutation more than once?

4. Make a guess. How many permutations of a set of 6 cards do you think there are? (Be sure to include the permutation [1,2,3,4,5,6]) We'll figure out how to count them later, just make a guess here.

A new way to think about permutations

In the previous questions, we were talking about permutations as things. By permutation, we meant a certain order of the elements in our set. Now, we are going to try to think about a permutation as a set of instructions about *how* to order a set. In this section, you will try to put the cards in order as instructed.

Put your cards on the table in front of you, face up, in this order [2,5,6,3,1,4], left to right. This is sequence notation, and it tells us directly where to put each card. Another way to write this permutation is (1,5,2)(3,4,6). This notation we will call ***Cycle Notation*** and we'll use parentheses rather than brackets to help distinguish between the two notations. The expression (1,5,2)(3,4,6) means that the 1 goes to the fifth position, the 5 goes to the second position, and the 2 goes to the first. It also means that the 3 goes in the fourth position, the 4 in the sixth, and the 6 in the third.

5. Which of the permutations below are the same? Draw a line matching a permutation in sequence notation to the same permutation in cycle notation.

Sequence Notation	Cycle Notation
$[3,4,5,6,1,2]$	(1,3,5)(2,4,6)
$[2,6,1,5,4,3]$	(1,3,6,2)(4,5)
$[6,5,4,3,2,1]$	(1,5,3)(4,2,6)
$[5,6,1,2,3,4]$	(1,6)(2,5)(3,4)

6. Write the permutations (2,5)(3,4,1,6) and (3,6,5)(1,2,4) in sequence notation.

7. Write the permutations [2,4,1,5,6,3] and [6,3,5,1,2,4] in sequence notation.

Sometimes a card should be directed to go to "its own" position. For instance, in the permutation [3,4,2,1,5,6], the 5 and 6 are already in the position they're supposed to be in. When this happens, we write them on their own, not in parentheses. That permutation looks like (1,4,2,3) 5 6 when written in cycle notation.

8. Write the permutations [5,3,2,4,1,6] and [3,6,5,4,2,1] in cycle notation.

9. Write the permutations 3 (6,2,1)(5,4) and (3,1) 4 (5,2) 6 in sequence notation.

10. Shuffle your deck six times. Use the permutations you get to fill in the following table.

Sequence Notation	Cycle Notation

Activity 1 Permutations

May 2010 Cliff Smith

Name: SOLUTIONS

What is a permutation?

Informally, a *permutation* of a set is a re-ordering of that set. Shuffling a deck of cards offers a good example of a permutation. There are many different ways to talk about and write permutations. We will explore a few of them here. Start with the deck of cards your teacher gave you. Put the deck in ascending numeric order. When you draw from the top of the deck, the first card you draw should be 1, followed by 2, and so on. This permutation is going to be our starting point. We will write permutations in relation to this order as well as in relation to each other.

1. Shuffle the deck a few times, then draw the cards one at a time from the top of the deck. Use the table below to keep track of the order in which you draw the cards.

1st card	2nd card	3rd card	4th card	5th card	6th card
6	1	5	2	4	3

2. The order in which you wrote the cards in the previous problem is one way to write a permutation. For clarity, we will call writing permutations in this way ***Sequence Notation***. Simply write the numbers in the order in which they appear. Do it twice more here on your own. Shuffle the deck, then draw the cards once at a time and write them in the order they appear. Put all the numbers into one set of [brackets]

[6, 3, 4, 5, 1, 2]

[3, 4, 2, 5, 6, 1]

3. Did you get the same permutation more than once? NO

- IT IS NOT LIKELY THAT STUDENTS WILL OBTAIN THE SAME PERMUTATION

4. Make a guess. How many permutations of a set of 6 cards do you think there are? (Be sure to include the permutation [1,2,3,4,5,6]) We'll figure out how to count them later, just make a guess here.

- THIS QUESTION IS DESIGNED TO GUIDE STUDENT ATTENTION TO THE IDEA THAT THERE ARE ONLY A CERTAIN NUMBER of WAYS TO PERMUTE A SET.

- YOU MAY LEARN SOMETHING ABOUT STUDENT THOUGHT BY EXAMINING ANSWERS TO THIS QUESTION

A new way to think about Permutations

In the previous questions, we were talking about permutations as things. By permutation, we meant a certain order of the elements in our set. Now, we are going to try to think about a permutation as a set of instructions about *how* to order a set. In this section, you will try to put the cards in order as instructed.

Put your cards on the table in front of you, face up, in this order [2,5,6,3,1,4], left to right. This is sequence notation, and it tells us directly where to put each card. Another way to write this permutation is (1,5,2)(3,4,6). This notation we will call ***Cycle Notation*** and we'll use parentheses rather than brackets to help distinguish between the two notations. The expression (1,5,2)(3,4,6) means that the 1 goes to the fifth position, the 5 goes to the second position, and the 2 goes to the first. It also means that the 3 goes in the fourth position, the 4 in the sixth, and the 6 in the third.

5. Which of the permutations below are the same? Draw a line matching a permutation in sequence notation to the same permutation in cycle notation.

Sequence Notation	Cycle Notation
[3, 4, 5, 6, 1, 2]	(1,3,5)(2,4,6)
[2, 6, 1, 5, 4, 3]	(1,3,6,2)(4,5)
[6, 5, 4, 3, 2, 1]	(1,5,3)(4,2,6)
[5, 6, 1, 2, 3, 4]	(1,6)(2,5)(3,4)

6. Write the permutations (2,5)(3,4,1,6) and (3,6,5)(1,2,4) in sequence notation.

(2,5)(3,4,1,6)

⇓

[4,5,6,3,2,1]

(3,6,5)(1,2,4)

⇓

[4,1,5,2,6,3]

7. Write the permutations [2,4,1,5,6,3] and [6,3,5,1,2,4] in sequence notation.

[2,4,1,5,6,3]

⇓

(1,3,6,5,4,2)

[6,3,5,1,2,4]

⇓

(5,3,2)(4,6,1)

Sometimes a card should be directed to go to "its own" position. For instance, in the permutation [3,4,2,1,5,6], the 5 and 6 are already in the position they're supposed to be in. When this happens, we write them on their own, not in parentheses. That permutation looks like (1,4,2,3) 5 6 when written in cycle notation.

8. Write the permutations [5,3,2,4,1,6] and [3,6,5,4,2,1] in cycle notation.

[5,3,2,4,1,6]

⇓

(2,3)(1,5),4,6

[3,6,5,4,2,1]

⇓

(5,3,1,6,2) 4

9. Write the permutations 3 (6,2,1)(5,4) and (3,1) 4 (5,2) 6 in sequence notation.

3 (6,2,1)(5,4)

⇓

[2,6,3,5,4,1]

(3,1) 4 (5,2) 6

⇓

[3,5,1,4,2,6]

10. Shuffle your deck six times. Use the permutations you get to fill in the following table.

Sequence Notation	Cycle Notation
[1,4,5,6,3,2]	(2,6,4) 1 (5,3)
[4,1,5,2,6,3]	(2,4,1)(5,3,6)
[2,5,4,1,3,6]	(5,2,1,4,3) 6
[4,6,1,2,3,5]	(1,3,5,6,2,4)
[2,1,6,3,5,4]	(2,1)(6,3,4) 5
[5,2,1,3,6,4]	(1,3,4,6,5) 2

NOTE: PERMUTATIONS IN CYCLE NOTATION CAN BE WRITTEN MULTIPLE WAYS. FOR EXAMPLE, THE TWO PERMUTATIONS, (2,4,1)(5,3,6) AND (3,6,5)(1,2,4) ARE EQUIVALENT.

9 Lesson 2: Permutations as functions

Lesson 2 is a preliminary to the introduction of the Fibonacci solitaire algorithm which occurs in Lesson 4.

Continuing and broadening students' understanding of permutation is the goal of the next lesson. Specifically, this lesson is designed to incorporate a functional view of permutation into the students existing knowledge. Students will learn to repeatedly apply a single permutation to achieve differing results. I mean this lesson to guide students' thinking of a permutation from simply an order of a set, to a process that orders a set in a certain way and can be repeated or combined with other processes. The goal is to develop in the student the understanding of permutations as functions. The purpose of this lesson is not to lead students toward being able to understand Fibonacci solitaire, but only to strengthen the concept of permutation.

Notes to the instructor

Materials *Cards:* Students will need a deck of numbered cards. A deck of size 10 is definitely sufficient. Many of the subsequent lessons require only six cards. A deck of standard playing cards, partitioned into suits, with King, Queen, and Jack removed will give 4 sets of 10 cards when the Ace is considered as a 1. If playing cards, or simply numbered cards are unavailable, you can have students make their own cards using index cards or slips of paper.

Lesson worksheets: The activity below can be photocopied and should be distributed to students. An overhead can be made from the activity as well, if so desired, then students will work with their own paper to answer the questions projected.

Goals Students develop understanding of permutation as a function

Prerequisite student knowledge Students should know and understand the definition of function.

Students should be able to determine whether a given relation, presented in non-symbolic and non-graphical form, is a function

Possible student misconceptions Students will repeatedly apply a permutation to itself and to other permutations. Because students will be obtaining different outputs each time they apply a permutation to itself, they may view this as evidence that permutations are not functions. That is, students may feel that a permutation has multiple outputs without considering the fact that they are actually considering the result of multiple inputs. The instructor should direct the students to consider what information they are using to determine the 'output' from the permutation. Direct students' attention to the initial order of the objects in the set as the 'input' to the permutation.

Students may have difficulties understanding how to permute objects from a previously permuted set. The use of numbered cards may indicate to the student that there is only one initial order on the set. These students may benefit from using objects that do not have an intrinsic order. Ask this student to use objects such as a pencil, some keys, an eraser, a card, a coin and any other objects that are easy to find and distinct. They should then place them in any order on the table and use the permutation as instructions about which item should go where when permuted. It is important for students to understand that the numbers in a permutation can be taken to mean 'the object in that position' rather than 'this number'.

Mathematics Common Core Standards 2010 A-IF-1: Understand that a function from one set (called the domain) to another set (called the range) assigns to each element of the domain exactly one element of the range. If f is a function and x is an element of its domain, then f(x) denotes the output of f corresponding to the input x.

Activity 2 Permutations

May 2010 Cliff Smith

Name: ______________________________________

How to permute permutations

Last time we learned how to write permutations two different ways. Remind yourself how to do it below.

1. Shuffle the cards and deal them out face up placing each card to the right of those before. Write the resulting permutation in both Sequence and Cycle notations.

We said that Cycle notation could be thought of as instructions about how to permute your set. We'll make that more clear now.

2. Place your cards in order on the table in front of you according to the permutation (4,2,3) (1,6) 5. Write it in sequence notation.

3. Use that same permutation, (4,2,3) (1,6) 5, as instructions about how to permute the cards again. For instance, the 4 went to the second position the first time. Now move the card that is in the fourth position to the second position. The permutation tells you to take the card that's in the second position, and put it in the third. It also says to move the card that's in the third position to the fourth position. The cards in the positions designate by the first set of parentheses all swapped places. Continue this process. Write the result of this second application of the permutation in both Sequence and Cycle notation.

4. Apply the same permutation to the new order. The card that's in the fourth position (not necessarily 4) will move to the second position and so on. Write the new permutation in both sequence and cycle notations.

Do it some more

5. Shuffle your deck and write the resulting permutation in cycle notation here.

6. Use the permutation you got in the last problem and apply it repeatedly like in the previous section.

# of times permutation applied	Resulting permutation

7. Essay/Long answer: If you continue applying a permutation to itself like this, will it eventually get back to the original order of the set? Why or why not?

Activity 2 Permutations

May 2010 Cliff Smith

Name: SOLUTIONS

How to permute permutations

Last time we learned how to write permutations two different ways. Remind yourself how to do it below.

1. Shuffle the cards and deal them out face up placing each card to the right of those before. Write the resulting permutation in both Sequence and Cycle notations.

[6, 2, 3, 5, 4, 1] ⟹ (1, 6)(5, 4), 2, 3

We said that Cycle notation could be thought of as instructions about how to permute your set. We'll make that more clear now.

2. Place your cards in order on the table in front of you according to the permutation (4,2,3) (1,6) 5. Write it in sequence notation.

[6, 4, 2, 3, 5, 1]

3. Use that same permutation, (4,2,3) (1,6) 5, as instructions about how to permute the cards again. For instance, the 4 went to the second position the first time. Now move the card that is in the fourth position to the second position. The permutation tells you to take the card that's in the second position, and put it in the third. It also says to move the card that's in the third position to the fourth position. The cards in the positions designate by the first set of parentheses all swapped places. Continue this process. Write the result of this second application of the permutation in both Sequence and Cycle notation.

[1, 3, 4, 2, 5, 6]

4. Apply the same permutation to the new order. The card that's in the fourth position (not necessarily 4) will move to the second position and so on. Write the new permutation in both sequence and cycle notations.

[6, 2, 3, 4, 5, 1]

Do it some more

5. Shuffle your deck and write the resulting permutation in cycle notation here.

[2, 6, 3, 1, 4, 5]

⇓

(2, 1, 4, 5, 6) 3

6. Use the permutation you got in the last problem and apply it repeatedly like in the previous section. (2,1,4,5,6) 3

# of times permutation applied	Resulting permutation
1	[2,6,3,1,4,5]
2	[6,5,3, 2,1,4]
3	[5,4,3,6,2,1]
4	[4,1,3,5,6,2]
5	[1,2,3,4,5,6]
6	[2,6,3,1,4,5]
7	[6,5,3,2,1,4]

7. Essay/Long answer: If you continue applying a permutation to itself like this, will it eventually get back to the original order of the set? Why or why not?

– THE TABLE ABOVE WAS CONSTRUCTED OF SUFFICIENT LENGTH SO, IF A STUDENT PERFORMS THE PERMUTATIONS CORRECTLY, THE STUDENT WILL OBTAIN THE IDENTITY PERMUTATION. THE HOPE IS THAT STUDENTS WILL START TO UNDERSTAND THE CYCLIC NATURE OF PERMUTATIONS.

10 Lesson 3: Counting permutations

Lesson 3 is a preliminary to the introduction of the Fibonacci solitaire algorithm which occurs in Lesson 4.

An integral part of the Fibonacci solitaire lesson sequence is determination of the probability of winning. To do this successfully, students need to know how to count objects such as permutations. This lesson introduces the student to counting combinatorially, using the multiplicative counting rule, and guides them to develop the factorial for counting permutations. This introduction will help students achieve greater success when asked to count more difficult objects such as perfect pairings and riggings, both of which can be counted using multiplication.

Notes to the instructor

Materials *Cards:* Students will need a deck of numbered cards. A deck of size 10 is definitely sufficient. Many of the subsequent lessons require only six cards. A deck of standard playing cards, partitioned into suits, with King, Queen, and Jack removed will give 4 sets of 10 cards when the Ace is considered as a 1. If playing cards, or simply numbered cards are unavailable, you can have students make their own cards using index cards or slips of paper.

Lesson worksheets: The activity below can be photocopied and should be distributed to students. An overhead can be made from the activity as well, if so desired, then students will work with their own paper to answer the questions projected.

Goals Students will learn simple combinatorial counting techniques.

Students will count the number of permutations of a given set.

Students will encounter and understand factorial notation.

Students will evaluate factorials by hand and with technology.

Prerequisite student knowledge Students should be at a high school mathematics level.

Possible student misconceptions Students may fail to develop, in the first few problems, the understanding that these numbers can be calculated with multiplication, but instead may list out all possibilities and count. For such a student, the instructor may want to help the student organize their lists so that it becomes clear that for each of a certain attribute, there are a constant number of possibilities. Remind this student that this form of repeated addition is, in fact, a simple multiplication problem.

Students may falter when presented with factorial notation. Some may ask, “Why?”. The instructor might ask in return “why do we use + when we mean add?” The answer is that sometimes we have to do something quite often and it behooves us to save time and paper by writing $n!$ rather than the entire multiplication problem.

Activity 3 Permutations

May 2010 Cliff Smith

Name: ____

Counting your options

1. A restaurant has 3 types of meat and 2 types of bread. How many different sandwiches can they make?

2. You are going to buy a car. There are 3 paint colors to choose from and 3 types of upholstery. How many different ways can you order your car?

3. Mr. Rogers has 7 cardigan sweaters, 9 pairs of loafers, and 4 pairs of pants. How many different outfits can he put together?

In all of the above problems, you were told how many choices you had for each item. If you multiplied these choices together you would get the total number of ways to choose your options. For example, if you chose "mystery meat #1" you could then get it two ways, with one bread or the other. If you choose "mystery meat #2" you can get that sandwich two ways also. Same with the third meat option. You are adding 2 to itself 3 times, ie multiplying 3 and 2.

4. Write some of your friends' names here. Count how many you wrote.

5. Write some of the ways you could get home after school (if it was ok with your parents). How many did you write?

6. Write some things you might do when you get home. How many?

7. Assume you go home with one friend, and only do one activity when you get home. How many ways are there for you plan your afternoon?

How many permutations are there?

8. Pretend you've shuffled the same deck of cards we've been using previously. How many options are there for the first card?

9. Now that the first card has been drawn, how many remaining possibilities are there for the second card?

10. How many for the third card?

11. Keeping that in mind, write an expression that will give the number of permutations of six objects.

12. Write an expression that will give the number of permutations of 10 objects.

This is called a ***factorial***, and it is written with an exclamation point, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$. Your calculator should have an exclamation point and if you use it, it will do the math for you.

13. Use your calculator to find 20! and 30!.

14. Use your calculator to find the number of ways to permute a set of 40 objects.

Activity 3 Permutations

May 2010 Cliff Smith

Name: SOLUTIONS

Counting your options.

1. A restaurant has 3 types of meat and 2 types of bread. How many different sandwiches can they make?

$3 \cdot 2 = \boxed{6}$

MEAT 1 - BREAD 1
MEAT 1 - BREAD 2
MEAT 2 - BREAD 1
MEAT 2 - BREAD 2
MEAT 3 - BREAD 1
MEAT 3 - BREAD 2

2. You are going to buy a car. There are 3 paint colors to choose from and 3 types of upholstery. How many different ways can you order your car?

$3 \cdot 3 = \boxed{9}$

3. Mr. Rogers has 7 cardigan sweaters, 9 pairs of loafers, and 4 pairs of pants. How many different outfits can he put together?

$7 \cdot 9 \cdot 4 = 252$

- THE FIRST TWO PROBLEMS CAN BE COUNTED BY LISTING COMBINATIONS. THE HOPE IS THAT STUDENTS, WHEN FACED WITH THE ENORMITY OF PROBLEM 3, WILL SEARCH FOR RELATIONSHIPS BETWEEN THE OPTIONS AND TOTAL NUMBER. WE HOPE THEY WILL DISCOVER THE MULTIPLICATIVE NATURE AND WILL BE ABLE TO JUSTIFY SUCH MULTIPLICATION.

In all of the above problems, you were told how many choices you had for each item. If you multiplied these choices together you would get the total number of ways to choose your options. For example, if you chose "mystery meat #1" you could then get it two ways, with one bread or the other. If you choose "mystery meat #2" you can get that sandwich two ways also. Same with the third meat option. You are adding 2 to itself 3 times, ie multiplying 3 and 2.

4. Write some of your friends' names here. count how many you wrote.

DAWN DAKOTA JOAN
FAYE BRITTINA SPENCER
RICHARD LARRY

8

5. Write some of the ways you could get home after school (if it was ok with your parents). How many did you write?

SKATE TAXI TRAIN
BUS HELICOPTER CRAWL
BIKE JET-PACK HORSE
WALK PARENTS PICK ME UP

11

6. Write some things you might do when you get home. How many?

X-BOX CALL JOE
HOMEWORK PAINT
EAT
PLAY DRUMS

6

7. Assume you go home with one friend, and only do one activity when you get home. How many ways are there for you plan your afternoon?

$8 \cdot 11 \cdot 6 = 528$

How many permutations are there?

8. Pretend you've shuffled the same deck of cards we've been using previously. How many options are there for the first card?

6 OPTIONS $\{1, 2, 3, 4, 5, 6\}$

9. Now that that first card has been drawn, how many remaining options are there for the second card?

5 OPTIONS

10. How many for the third card?

4

11. Keeping that in mind, write an expression that will give the number of permutations of six objects.

$6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$

12. Write an expression that will give the number of permutations of 10 objects.

$10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 3{,}628{,}800$

This is called a ***factorial***, and it is written with an exclamation point, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$. Your calculator should have an exclamation point and if you use it, it will do the math for you.

13. Use your calculator to find 20! and 30!.

$20! = 2432902008176640000$

$30! = 265252859812191058636308480000000$

14. Use your calculator to find the number of ways to permute a set of 40 objects.

$40! \approx 8.16 \cdot 10^{47}$

- BIG NUMBERS CAN BE USED TO REMIND STUDENTS OF SCIENTIFIC NOTATION.
- ENCOURAGE STUDENTS TO EXPLORE MENUS ON THEIR CALCULATORS.

11 Lesson 4: Fibonacci solitaire

This lesson is designed mainly to help the student become comfortable with the algorithm and the associated notation. Additionally, this is the lesson where students work to decide whether the algorithm is a function. Understanding the algorithm as a function is key to understanding the counting arguments that come later in the lesson sequence. Students should be able to tell fairly quickly that the algorithm is sufficiently specific to limit the output to a unique pairing given any order of the cards in the deck. This lesson also hints at the fact that the algorithm is not injective. Students can work together to determine whether two different permutations lead to the same resultant pairing. Formalization of the injective property, and the algorithm's lack of it will wait until the next lesson.

Notes to the instructor

Materials *Cards:* Students will need a deck of numbered cards. A deck of size 10 is definitely sufficient. Many of the subsequent lessons require only six cards. A deck of standard playing cards, partitioned into suits, with King, Queen, and Jack removed will give 4 sets of 10 cards when the Ace is considered as a 1. If playing cards, or simply numbered cards are unavailable, you can have students make their own cards using index cards or slips of paper.

Algorithm instruction sheet: The algorithm is called Fibonacci solitaire and the sheet with instructions and an example game is at the end of this lesson. Students will be instructed to ask you for the instruction sheet when they are ready. If you wish, you may give it to them with the rest of their lesson worksheets.

Lesson worksheets: The activity below can be photocopied and should be distributed to students. An overhead can be made from the activity as well, if so desired, then students will work with their own paper to answer the questions projected.

Goals Students will follow algorithmic instructions.

Students will determine whether an algorithm is a function.

Students will analyze domain and range of the algorithm using previous knowledge about permutations.

Students will be introduced to the injective concept, if not the vocabulary.

Prerequisite student knowledge Students should know and understand the function definition.

Students should understand the concepts of domain and range.

Possible student misconceptions Students may feel that the algorithm is not a function because it does not have an equation, or because it cannot be graphed. They may feel it is not a function because they cannot apply the vertical line test to it. These are common misconceptions about function that often arise because the majority of examples in the classroom are given in such a context. Students should be referred to the *definition* of function. This misconception may need to be addressed directly.

This function is very different from functions that a student might have previous experience with. Particularly, the function has the set of permutations as a domain rather than the set of real numbers. Students' previous work with real functions, and the fact that cards are drawn one at a time may lead the students to believe that the algorithm has the set of integers as domain. Direct the students' attention to the start of the game and the end of the game. What do they start the game with? A shuffled set of cards, or a permutation. What does the game give at the end? A set of pairs and singles (a pairing). Directing their attention to the entirety of the process rather than a single step may help them to make the connection to permutation that we desire.

Students may initially show confusion about the standard notation we require. It is difficult to explain in writing how to organize the pairing. Direct student's attention to the example game and how its result is written. Perhaps ask the students to organize their cards as in the example game, and play. What do they obtain as the final result? How would they have to change their result to make it align with the example? Do not allow this to take up too much time. The point of the lesson is the justification of the algorithm as a function. If necessary, give the student direct instruction about how to write some of their results.

It may happen that no two students in the classroom obtain the same final pairing. If students are apt to think that every result comes from a unique permutation, then the instructor should step in and correct this logical misstep. Offer the students two example permutations that will give the same resulting pairing. Ask them to apply the algorithm to both of your permutations, they should see, that some pairings are the result of more than one permutation. As an example, consider a set of four cards. If the initial permutation of these cards is $[4, 2, 3, 1]$ in that order, then the result will be $1, (2, 3), 4$. Now, play the game again starting with the permutation $[2, 3, 4, 1]$. You will again end at $1, (2, 3), 4$. Tell the students to search together for an example using their 8-card deck.

Fibonacci solitaire

To play the game, you will need a deck with an even number of cards. The cards should be numbered 1, 2, 3,... and so on.

Directions for gameplay

Shuffle the deck. Draw a card from the top of the deck and place it face up on the table in front of you. Draw the next card from the top of the deck and compare it to those on the table. The following rules dictate what to do with this card.

FS 1 If the drawn card is **smaller** in value than the smallest card on the table, place it to the left of any cards on the table.

FS 2: If the drawn card is **larger** in value than the smallest card on the table, pair it with the smallest card on the table, removing them both from the table and placing this pair aside.

If at any point the table is empty, simply place the drawn card face up on the table and continue playing according to the previous rules. When all the cards have been played, the game is over. There may be cards remaining on the table. If no cards remain on the table, then all cards have been paired, and you win.

We will often have need to refer back to these original rules. When we do so, we will refer to them as **FS 1** and **FS 2**.

An example game

Here we will play an example game together using a six-card deck. It may be helpful to organize your deck as indicated and play through the example with your cards. Let us suppose that the cards in the deck have the following order: 4, 6, 5, 1, 2, 3. The table below shows which card is drawn, and how the game should be played, given that card.

Card drawn	On the table	Removed from play
4	4	
6		(4,6)
5	5	(4,6)
1	1 5	(4,6)
2	5	(4,6) (1,2)
3	3 5	(4,6) (1,2)

Since there are two cards left on the table, this game does not result in a win. The initial permutation of the cards was [4, 6, 5, 1, 2, 3] and the final pairing is (1,2), 3, (4,6), 5

Activity 4 Fibonacci Solitaire

May 2010 Cliff Smith

Name: ____

1. The instruction sheet your teacher gave you tells how to play a game called Fibonacci Solitaire. Follow the instructions and play a few games with a deck of 8 cards, keeping track of the order in which you drew the cards as well as the resulting pairings. Always write the pairs with the smaller number first. Write the results with the singles and the small cards from each pair organized numerically, in ascending order. See the example below, the numbers in the small boxes are the values of cards in the order they were drawn the resulting set of pairs and singles is written under "Final pairing". Ace is low.

1st	2nd	3rd	4th	5th	6th	7th	8th	Final pairing
6	3	8	7	4	2	1	5	(1,5), 2, (3,8), 4, (6,7)

2. If you play the game several times with the same initial order of the cards in your deck, will you always end up with the same result? Try it.

3. Is it possible for a single permutation of the deck to lead to two different pairings?

4. Does this algorithm define a function? Provide justification for your answer. (examples don't count as justification)

Most functions you've seen up to this point take the set of real numbers as a domain, and usually have a subset of the real numbers as the range. This function is a bit different.

5. What is the domain of this algorithm?

6. Describe the range of this algorithm.

7. Compare your results with those of your classmates. Do any of your classmates have a result the same as any of yours? If so, did they have the same permutation of their deck of cards as you did?

8. Find two permutations that are different, but that lead to the same final result.

Activity 4 Fibonacci Solitaire

May 2010 Cliff Smith

Name: SOLUTIONS

1. The instruction sheet your teacher gave you tells how to play a game called Fibonacci Solitaire. Follow the instructions and play a few games with a deck of 8 cards, keeping track of the order in which you drew the cards as well as the resulting pairings. Always write the pairs with the smaller number first. Write the results with the singles and the small cards from each pair organized numerically, in ascending order. See the example below, the numbers in the small boxes are the values of cards in the order they were drawn the resulting set of pairs and singles is written under "Final pairing". Ace is low.

1st	2nd	3rd	4th	5th	6th	7th	8th	Final pairing
6	3	8	7	4	2	1	5	(1,5), 2, (3,8), 4, (6,7)
4	7	2	3	8	1	5	6	(1,5)(2,3)(4,7) 6, 8
1	5	3	6	8	7	2	4	(1,5)(2,4)(3,6) 7, 8
4	8	6	1	2	5	3	7	(1,2)(3,7)(4,8) 5, 6
8	6	7	3	4	1	2	5	(1,2)(3,4) 5 (6,7) 8
2	6	3	8	5	4	1	7	(1,7) (2,6) (3,8) 4, 5
3	8	6	2	7	1	4	5	(1,4) (2,7) (3,8) 5, 6

2. If you play the game several times with the same initial order of the cards in your deck, will you always end up with the same result? Try it.

YES. IF I PLAY WHEN CARDS IN ORDER [4, 7, 2, 3, 8, 1, 5, 6]

I ALWAYS GET (1, 5)(2, 3)(4, 7) 6, 8

3. Is it possible for a single permutation of the deck to lead to two different pairings? NO, EVERY TIME I DRAW A CARD I CAN ONLY DO ONE THING WITH IT.

4. Does this algorithm define a function? Provide justification for your answer. (examples don't count as justification) YES. EACH ORDER OF THE CARDS CAN ONLY END AT ONE RESULTING PAIRING. THE RULES ONLY ALLOW ONE THING TO HAPPEN AT EACH STEP.

Most functions you've seen up to this point take the set of real numbers as a domain, and usually have a subset of the real numbers as the range. This function is a bit different.

5. What is the domain of this algorithm?

ORDERS OF A DECK OF CARDS

-OR- PERMUTATIONS OF THE NUMBERS 1-8

6. Describe the range of this algorithm.

EVERYTHING IN THE RANGE IS MADE OF PAIRS AND SINGLE CARDS.

7. Compare your results with those of your classmates. Do any of your classmates have a result the same as any of yours? If so, did they have the same permutation of their deck of cards as you did?

MARK AND CRAIG HAD THE SAME RESULTS, BUT NOBODY HAD ANY THE SAME AS MINE.

8. Find two permutations that are different, but that lead to the same final result.

[1,5,3,6,8,7,2,4] AND [8,7,3,6,2,4,1,5]

BOTH OF THESE ORDERS OF THE DECK GIVE THE RESULT

(1,5)(2,4)(3,6) 7, 8

12 Lesson 5: Experimental probability

Determining the theoretic probability of a winning game of Fibonacci solitaire is the motivating force behind all of the work that students will do with permutations, rigging, and inverse functions. It is not necessary, however to wait until the theoretic probability is determined before developing experimental probabilities with which we can compare the theoretic determination later. This lesson guides students through a number of games with different deck sizes and asks them to calculate experimental probabilities by obtaining sample information from peers. Students will make a conjecture about how the probability of the algorithm behaves as deck size increases.

Notes to the instructor

Materials *Cards:* Students will need a deck of numbered cards. A deck of size 10 is definitely sufficient. Many of the subsequent lessons require only six cards. A deck of standard playing cards, partitioned into suits, with King, Queen, and Jack removed will give 4 sets of 10 cards when the Ace is considered as a 1. If playing cards, or simply numbered cards are unavailable, you can have students make their own cards using index cards or slips of paper.

Algorithm instruction sheet: The algorithm is called Fibonacci solitaire and the sheet with instructions is at the end of the previous lesson. Students should have kept their instruction sheets from the prior lesson, but you may find it handy to have a few extra copies available.

Lesson worksheets: The activity below can be photocopied and should be distributed to students. An overhead can be made from the activity as well, if so desired, then students will work with their own paper to answer the questions projected.

Goals Students will cooperate to find an experimental probability for winning the Fibonacci solitaire game with two different deck sizes.

Students will make a conjecture about the theoretical probability based on their examples.

Prerequisite student knowledge Students should have completed lesson 4 and be familiar with the Fibonacci solitaire algorithm.

Students must know what a permutation is and how to identify it.

Students should be familiar with simple probability and how to find it.

Possible student misconceptions Students may feel that the theoretic probability is the *real* probability, and that their experimental probability is wrong, or is just an approximation to the *real* probability. Students may also hold these views in the reverse order. This is a good lesson to help the students understand that these two different probabilities, obtained in different ways, are equally valid, and have differing uses. Many probabilities in real contexts cannot be calculated, but must be determined experimentally. Probabilities determined in this way are not wrong. Students should know that experimental probability will, statistically speaking, approach the theoretic probability as the sample size increases.

Students may obtain different experimental probabilities and view this as meaning that they are wrong. This should be discouraged. Students often want their answers and work to be the same as others, this mirrors the view that there is only one way to do mathematics, and that any other way is incorrect. This should also be discouraged.

Activity 5 Fibonacci Solitaire

May 2010 Cliff Smith

Name: ____

1. Using a "deck" with only two cards, play the Fibonacci solitaire game 5 times. When you are finished, count how many times you won and find your experimental probability of winning. Divide the number of games you won by the total number of games you played.

Initial permutation	Final pairing

2. Ask 4 of your classmates how many times they won when they played with only two cards. Using this sample, determine the experimental probability of winning. Divide the total number of wins you and your four classmates had by the total number of games you and your four classmates played.

3. How many different ways are there to order the cards in a two-card deck? List them.

4. How many of the permutations you found in the previous problem result in a win?

5. Find the theoretic probability of winning with a two-card deck. Divide the number of orderings of the deck that result in a win by the total number of orders of the deck.

6. Compare the theoretic probability from the last problem with the experimental probabilities you calculated in problems 1 and 2. How do they compare? Which of the experimental probabilities is closer to the theoretic probability you found in the last problem?

7. Now you need a deck of four cards. Play the game 5 times. Find your experimental probability for these 5 games.

Initial permutation	Final pairing

8. Ask 4 of your classmates how many times they won when they played with a four-card deck. Combined with your results, you now have a sample size of 25 games. Determine the experimental probability of winning with this sample.

9. Develop and organize a table that contains all the permutations of a four-card deck, (remember that 4! is the number of permutations). Determine whether or not each game ends in a win. Find the theoretic probability of winning with a four-card deck.

10. Examine the two experimental probabilities you found, and the theoretic probability you developed. Which of the experimental probabilities is closer to the theoretic? any ideas why?

11. Play the game 5 times with an eight-card deck. Find your experimental probability with this sample of five.

Initial permutation	Final pairing

12. Ask 4 of your classmates how many times they won when they played with a four-card deck. Combined with your results, you now have a sample size of 25 games. Determine the experimental probability of winning with this sample.

13. Which of these two experimental probabilities do you think is closer to the theoretic probability of winning with an eight-card deck? Why do you think so?

14. Make a conjecture about the experimental probability you would obtain if you used the results of all of your classmates? How big would the sample size be?

Activity 5 Fibonacci Solitaire

May 2010 Cliff Smith

Name: SOLUTIONS

1. Using a "deck" with only two cards, play the Fibonacci solitaire game 5 times. When you are finished, count how many times you won and find your experimental probability of winning. Divide the number of games you won by the total number of games you played.

Initial permutation	Final pairing
[2, 1]	1, 2
[2, 1]	1, 2
[2, 1]	1, 2
[2, 1]	1, 2
[1, 2]	(1, 2)

$\frac{4}{5} = .8$

2. Ask 4 of your classmates how many times they won when they played with only two cards. Using this sample, determine the experimental probability of winning. Divide the total number of wins you and your four classmates had by the total number of games you and your four classmates played.

CRAWFORD WON 2 TIMES

FRANK WON 4 TIMES

SUE WON 3 TIMES

ROSIE WON ONCE.

$4 + 2 + 4 + 3 + 1 = 14$

$\frac{14}{25} = .56$

3. How many different ways are there to order the cards in a two-card deck? List them. [1, 2]
[2, 1]   2 WAYS.

4. How many of the permutations you found in the previous problem result in a win? ONLY ONE OF THEM WINS, [1, 2] WINS. [2, 1] DOES NOT WIN.

5. Find the theoretic probability of winning with a two-card deck. Divide the number of orderings of the deck that result in a win by the total number of orders of the deck.

$\frac{1}{2} = .5$

6. Compare the theoretic probability from the last problem with the experimental probabilities you calculated in problems 1 and 2. How do they compare? Which of the experimental probabilities is closer to the theoretic probability you found in the last problem?

MY RESULTS BY THEMSELVES DO NOT GIVE A PROBABILITY THAT'S VERY CLOSE TO THE THEORETIC PROBABILITY. BUT WHEN I COMBINED MY RESULTS WITH OTHERS' THE EXPERIMENTAL PROBABILITY IS PRETTY CLOSE TO THE THEORETIC ONE.

7. Now you need a deck of four cards. Play the game 5 times. Find your experimental probability for these 5 games.

Initial permutation	Final pairing	
[4, 3, 1, 2]	(1, 2) 3, 4	
[3, 4, 1, 2]	(1, 2) (3, 4)	WIN!
[2, 3, 1, 4]	(1, 4) (2, 3)	WIN!
[2, 3, 4, 1]	1 (2, 3) 4	
[4, 3, 2, 1]	1, 2, 3, 4	

$\frac{2}{5} = .4$

8. Ask 4 of your classmates how many times they won when they played with a four-card deck. Combined with your results, you now have a sample size of 25 games. Determine the experimental probability of winning with this sample.

OSCAR WON 0 GAMES
OLIVE WON 3 GAMES
HARRIETT WON 2 GAMES
PACO WON 2 GAMES

$0+2+3+2+2 = 9$

$\frac{9}{25} = .36$

9. Develop and organize a table that contains all the permutations of a four-card deck, (remember that 4! is the number of permutations). Determine whether or not each game ends in a win. Find the theoretic probability of winning with a four-card deck.

1 FIRST	2 FIRST	3 FIRST	. . .
[1, 2, 3, 4]	[2, 1, 3, 4]	[3, 1, 4, 2]	
[1, 3, 2, 4]	[2, 1, 4, 3]	[3, 1, 2, 4]	. . . .
[1, 4, 3, 2]	[2, 3, 4, 1]	⋮	
[1, 2, 4, 3]	[2, 3, 1, 4]		
[1, 3, 4, 2]	[2, 4, 1, 3]		
[1, 4, 2, 3]	[2, 4, 3, 1]		

- A TABLE LIKE THIS ONE WILL HELP YOUR STUDENTS MAKE SURE THAT THEY'VE COUNTED THEM ALL.

- ENCOURAGE STUDENTS TO WORK TOGETHER. IF THEY SPLIT THE WORK LOAD AMONGST A TEAM, THE WORK WILL NOT BE SO ARDUOUS.

$\frac{9}{24} = .375$

10. Examine the two experimental probabilities you found, and the theoretic probability you developed. Which of the experimental probabilities is closer to the theoretic? any ideas why?

THE TWO EXPERIMENTAL PROBABILITIES ARE EQUALLY CLOSE TO THE THEORETIC, THOUGH IN OPPOSITE DIRECTIONS.

11. Play the game 5 times with an eight-card deck. Find your experimental probability with this sample of five.

Initial permutation	Final pairing
[4,7,2,3,8,1,5,6]	(1,5)(2,3)(4,7)6,8
[1,5,3,6,8,7,2,4]	(1,5)(2,4)(3,6)7,8
[4,8,6,1,2,5,3,7]	(1,2)(3,7)(4,8)5,6
[8,6,7,3,4,1,2,5]	(1,2)(3,4)5(6,7)8
[2,6,3,8,5,4,1,7]	(1,7)(2,6)(3,8)4,5

$\frac{0}{5} = 0$

12. Ask 4 of your classmates how many times they won when they played with a four-card deck. Combined with your results, you now have a sample size of 25 games. Determine the experimental probability of winning with this sample.

$0 + 1 + 2 + 3 + 0 = 6$

XAVIER WON 1 GAME
DARRYL WON TWICE
TONI WON THREE TIMES
NICOLE WON 0 GAMES.

$\frac{6}{25} = .24$

13. Which of these two experimental probabilities do you think is closer to the theoretic probability of winning with an eight-card deck? Why do you think so? I DON'T THINK THAT THE PROBABILITY IS ZERO. ALSO, WHEN I COMBINED MY INFO WITH OTHERS' IN PROBLEM 2, WE GOT AN EXPERIMENTAL PROBABILITY CLOSER TO THE THEORETIC ONE. THE PROBABILITY IN PROBLEM 12 IS PROBABLY CLOSER.

14. Make a conjecture about the experimental probability you would obtain if you used the results of all of your classmates? How big would the sample size be?

IT SEEMS LIKE THE MORE TIMES WE TRY IT THE CLOSER THE PROBABILITY IS. I BET THAT THE PROBABILITY WOULD BE PRETTY CLOSE IF WE USED EVERYBODY'S INFORMATION. THERE ARE 29 PEOPLE IN CLASS. EVERYBODY PLAYED FIVE GAMES. THAT'S $29 \cdot 5 = 145$ GAMES.

13 Lesson 6: Fibonacci solitaire as a function

The initial lesson involving the Fibonacci solitaire game hinted at the non-injective nature of this algorithm. This lesson introduces and helps explain the concept of injective functions and formalizes the knowledge that the algorithm is not injective. The algorithm provides a concrete example of why the injective property is necessary for a function to have an inverse. Students will see that because the function is not injective, if they try to reverse their steps, they will not be able to do so in a well defined way. This concrete example will help to demystify the inclusion of the injective property in the definition of inverse function.

Notes to the instructor

Materials *Cards:* Students will need a deck of numbered cards. A deck of size 10 is definitely sufficient. Many of the subsequent lessons require only six cards. A deck of standard playing cards, partitioned into suits, with King, Queen, and Jack removed will give 4 sets of 10 cards when the Ace is considered as a 1. If playing cards, or simply numbered cards are unavailable, you can have students make their own cards using index cards or slips of paper.

Algorithm instruction sheet: The algorithm is called Fibonacci solitaire and the sheet with instructions is at the end of the previous lesson. Students should have kept their instruction sheets from the prior lesson, but you may find it handy to have a few extra copies available.

Lesson worksheets: The activity below can be photocopied and should be distributed to students. An overhead can be made from the activity as well, if so desired, then students will work with their own paper to answer the questions projected.

Goals Students will observe and support their claim that the Fibonacci solitaire function is not injective.

Students will discover the importance of the injective property to inverse functions.

Prerequisite student knowledge Students should have completed lesson 4 and be familiar with the Fibonacci solitaire algorithm.

Students must know what a permutation is and how to identify it.

Students should have, at least, been introduced to the injective property of functions.

Students should have studied, at least briefly, the concept of function inverses. Students should know how to find the inverse of a linear function.

Possible student misconceptions The injective property is often confused for the function definition. There is good reason for this. The two properties are so similar because the injective property allows an inverse relation to be a function. To reduce confusion, the injective property is often stated "a function f is one-to-one if whenever $f(a) = f(b)$, then $a = b$". The similarity between the injective property and function definition is confusing, but it can also be very enlightening. Talk as a class about how the two are related. Expect your students to struggle here, but do not be alarmed.

Students may believe the function is not injective because it does not pass the horizontal line test. They may believe this because they cannot draw the function, so cannot apply the test. Conversely, some students may believe that the function is injective for similar reasons, every horizontal line *does not* intersect this function because in different spaces, one in real space, the other in the space of permutations. Tell the students that because this function does not have the real numbers as a domain, it cannot be drawn on the familiar coordinate axes. Because of this, they cannot use the horizontal line test. Refer students to the *definition* of the injective property.

Students may think that finding a permutation that gives a certain result indicates that the process is invertible. The instructor should guide these students to look for other permutations that give the same result. Additionally, the instructor could tell the student to compare answers with peers and see whether they found the same permutation. Ask the student what it means that there are two permutations that give the same result. How does the student know for sure which permutation the given result arises from?

Activity 6 Fibonacci Solitaire

May 2010 Cliff Smith

Name: ______________________________________

1. Work with a friend to find two different permutations that lead to the same result after playing Fibonacci solitaire. What are the permutations? What is the result?

2. Find two numbers that are different from each other, but whose squares are equal. In other words, find two numbers a and b so that $a^2 = b^2$.

A function is called one-to-one, or injective, if every value of the range comes from a unique value in the domain. Stated another way, a function f is one-to-one if whenever $f(a) = f(b)$, then $a = b$. For example, the function $f(x) = x^2$ is not one-to-one because $2^2 = (-2)^2$ but $2 \neq (-2)$.

3. Is the Fibonacci solitaire function one-to-one? Explain.

4. Suppose someone told you that they played Fibonacci solitaire and that they got the result (1,2), 3, (4,6), 5. Can you find out what their starting permutation was? If so, find it. If not, say why.

An inverse function is one that undoes the process of another. For instance if f is a function, and g is its inverse, then if you apply f to an element of its domain, and apply g to the result, the output from g is the element to which you originally applied f.

5. Let $f(x) = x + 7$ what is the function that will undo what f does?

6. Shuffle your deck of cards. Write the permutation in sequence and cycle notations. What is the permutation that undoes that permutation? That is, What is the permutation that will return the cards to the order [1,2,3,4,5,6,7,8]

7. Essay/Long answer: Why is it important that a function be one-to-one in order for it to have an inverse?

Activity 6 Fibonacci Solitaire

May 2010 Cliff Smith

Name: SOLUTIONS

1. Work with a friend to find two different permutations that lead to the same result after playing Fibonacci solitaire. What are the permutations? What is the result?

[1,5,3,6,8,7,2,4] AND [8,7,3,6,2,4,1,5]

BOTH END AT THE RESULT, (1,5)(2,4)(3,6) 7,8

2. Find two numbers that are different from each other, but whose squares are equal. In other words, find two numbers a and b so that $a^2 = b^2$.

$a = 2$
$b = -2$

$a^2 = 2^2 = 4 = (-2)^2 = b^2$

A function is called one-to-one, or injective, if every value of the range comes from a unique value in the domain. Stated another way, a function f is one-to-one if whenever $f(a) = f(b)$, then $a = b$. For example, the function $f(x) = x^2$ is not one-to-one because $2^2 = (-2)^2$ but $2 \neq (-2)$.

3. Is the Fibonacci solitaire function one-to-one? Explain.

NO, IN PROBLEM 1 WE FOUND TWO DIFFERENT PERMUTATIONS THAT HAVE THE SAME OUTCOME.

4. Suppose someone told you that they played Fibonacci solitaire and that they got the result (1,2), 3, (4,6), 5. Can you find out what their starting permutation was? If so, find it. If not, say why.

[1,2,5,4,6,3] AND [4,6,5,3,1,2] BOTH GIVE THAT RESULT. I CAN'T SAY WHICH ONE IT CAME FROM.

An inverse function is one that undoes the process of another. For instance if f is a function, and g is its inverse, then if you apply f to an element of its domain, and apply g to the result, the output from g is the element you applied f to.

5. Let $f(x) = x + 7$ what is the function that will undo what f does?

$$f^{-1}(x) = x - 7$$

6. Shuffle your deck of cards. Write the permutation in sequence and cycle notations. What is the permutation that undoes that permutation? That is, What is the permutation that will return the cards to the order [1,2,3,4,5,6,7,8]

[3,8,6,7,5,2,1,4]
⇓
(1,7,4,8,2,6,3)

1 WENT TO 7,
7 WENT TO 4,
⋮
etc

TO UNDO, WE WANT

7 GOES TO 1
4 GOES TO 7,
⋮
etc

⇒ (3,6,2,8,4,7,1)

7. Essay/Long answer: Why is it important that a function be one-to-one in order for it to have an inverse?

– AS IN EXAMPLE IN PROBLEM 4, IF YOU CAN'T REVERSE THE PROCESS AND ARRIVE AT A SINGLE ANSWER, THEN THE REVERSE PROCESS CANNOT BE PERFORMED WITH CONFIDENCE.

14 Lesson 7: Undoing Fibonacci solitaire

Lesson 7 introduces the *depth* function, rigging, and the algorithm that is the reversal of Fibonacci Solitaire. Students will work through some tasks that require them to calculate depths of cards in certain pairings, then they will play through the original algorithm, keeping track of depths as they go. Rigging will be explained and students will be required to find the rigging of their pairing, then they will follow the reversal algorithm. Students will learn that with the addition of rigging, Fibonacci solitaire is now an injective function, and they will have experience with the inverse. This will lead to the next lesson where students will learn what a bijective correspondence is and how it can be used to reach our goals.

Notes to the instructor

Materials *Cards:* Students will need a deck of numbered cards. A deck of size 10 is definitely sufficient. Many of the subsequent lessons require only six cards. A deck of standard playing cards, partitioned into suits, with King, Queen, and Jack removed will give 4 sets of 10 cards when the Ace is considered as a 1. If playing cards, or simply numbered cards are unavailable, you can have students make their own cards using index cards or slips of paper.

Algorithm instruction sheet: The algorithm is called Fibonacci solitaire and the sheet with instructions is at the end of lesson 4. Students should have kept their instruction sheets from the prior lesson, but you may find it handy to have a few extra copies available.

Reverse Algorithm instruction sheet: It's easy to refer to this algorithm as the Reverse algorithm. Students will need a copy of the instructions to refer to. The Reverse Algorithm instruction sheet is at the end of this lesson. It is important that this sheet be separate from the students' worksheets so that they can refer to it while working.

Lesson worksheets: The activity below can be photocopied and should be distributed to students. An overhead can be made from the activity as well, if so desired, then students will work with their own paper to answer the questions projected.

Goals Students apply a function to a permutation to obtain information about that permutation

Students will be introduced to a new algorithm.

Students will understand that the new algorithm is the inverse of the Fibonacci solitaire algorithm.

Students will examine domain and range of an inverse function.

Prerequisite student knowledge Students should understand the function definition.

Students should have knowledge of the injective property of functions.

Students should know what an inverse function is.

Students should understand the relation between the injective property and inverse function.

Possible student misconceptions Students will be tempted to modify the rigging on pairings as they play backward. This may be a result of how the δ function was introduced and discussed. Students will want to equate rigging and depth. This should be discouraged. Remind students that a depth can change, but once a rigging is established, it stays put.

Students may be uncomfortable with the $\delta(a) = b$ notation. Many secondary students are uncomfortable with function notation, but should be encouraged to continue to work at understanding it. Continued practice will improve their ability to use the symbolic notation. Remind students that $\delta(a)$ is just the number of cards written to the right of a. The δ function is deceptively simple. Students may want to use their calculators, or fill out an equation. Remind them that they need only count.

Reverse algorithm

As you remove cards from the table according to the algorithm below, place them facedown in a single pile.

Compare the current depth of all bottoms with their rigging.

Reverse 1 If no bottoms have a current depth matching their rigging, remove the smallest singleton.

Reverse 2 If any bottoms have a current depth matching their rigging, consider the smallest such bottom, call it b. If there are no singletons smaller than b, remove the top that is paired with b. Otherwise, remove the smallest singleton.

Activity 7 Fibonacci Solitaire

May 2010 Cliff Smith

Name: ______________________________________

Depth

1. Shuffle the deck. Play a game of Fibonacci Solitaire. Be sure to write down the initial permutation of the cards in your deck as well as the resulting set of pairs and singletons.

The *depth*,δ of a card is the number of cards written to the right of that card. For instance, in the result (1,2), 3, (4,6), 5, the depth of 1 is 5, so we write $\delta(1) = 5$.

2. Find the depths of all the cards in your result from problem 1.

We have to keep track of our games in a different way now. As you play, you need to write down the *partial* results. For instance, if the first card you draw is a 4, the first partial result is 4. If the next card is a 6, then the next partial result is (4,6). Remember to write the partial results in numeric order. Below is an example from the permutation [4,6,5,1,2,3]

$$4$$

$$(4,6)$$

$$(4,6),5$$

$$1,(4,6),5$$

$$(1,2),(4,6),5$$

$$(1,2),3,(4,6),5$$

3. Shuffle your deck and play a game of Fibonacci solitaire. Keep track of your partial results like in the example above.

4. Find the depth of the lower card in each pair for all your partial results in the previous problem. If a card hasn't been drawn for a given partial result, then its depth is undefined. In the example above, $\delta(1)$ is undefined until the fourth step.

Rigging

What we really need to keep track of though is the depth of the lower cards in each pair when that pair is first created. In the example above, $\delta(4) = 1$ when the pair (4,6) was first made. Also $\delta(1) = 4$ when the pair (1,2) was first made. We will write this important information as a subscript, so that our new final result looks like $(1_4, 2), 3, (4_1, 6), 5$

5. Using your game from problem 3, determine the depth of the low card in each pair when that pair was first made. Write that information as a subscript on your final result.

6. Shuffle your deck and play Fibonacci solitaire. Write down all the partial results as well as the starting permutation, and find the depth of the lower card from each pair when that pair was first created.

This information that you are writing as a subscript on your final results we will call the *rigging*. The rigging will allow us to play the game backward.

7. Ask your instructor for a backward playing instruction sheet. Did you get one?

8. Follow the directions carefully and, starting from your final result in problem 6, play backward. Don't peek at your partial results from 6, just follow the directions and when you are done, check to see that you ended at your starting permutation from problem 6. If you ended elsewhere, make sure your rigging is correct and try again. Follow the instructions carefully.

9. Use the rigged pairing

$$1, (2_4, 6), (3_2, 10), 4, (5_3, 7), 8, 9$$

and play backward. Write the order of the deck you obtained as a result.

10. Remember what we said last time about inverse functions and one-to-one functions. Because we can now undo the Fibonacci solitaire function, what does that tell you?

11. What is the domain of the Fibonacci solitaire function?

12. What is the new range of the Fibonacci solitaire function?

Activity 7 Fibonacci Solitaire

May 2010 Cliff Smith

Name: SOLUTIONS

Depth

1. Shuffle the deck. Play a game of Fibonacci Solitaire. Be sure to write down the initial permutation of the cards in your deck as well as the resulting set of pairs and singletons.

[4, 7, 2, 3, 8, 1, 5, 6] gives result (1,5)(2,3)(4,7) 6, 8

The *depth*,δ of a card is the number of cards written to the right of that card. For instance, in the result (1,2), 3, (4,6), 5, the depth of 1 is 5, so we write $\delta(1) = 5$.

2. Find the depths of all the cards in your result from problem 1.

$\delta(1) = 7$

$\delta(5) = 6$

$\delta(2) = 5$

$\delta(3) = 4$

$\delta(4) = 3$

$\delta(6) = 1$

$\delta(8) = 0$

We have to keep track of our games in a different way now. As you play, you need to write down the *partial* results. For instance, if the first card you draw is a 4, the first partial result is 4. If the next card is a 6, then the next partial result is (4,6). Remember to write the partial results in numeric order. Below is an example from the permutation [4,6,5,1,2,3]

$$4$$
$$(4,6)$$
$$(4,6),5$$
$$1,(4,6),5$$
$$(1,2),(4,6),5$$
$$(1,2),3,(4,6),5$$

3. Shuffle your deck and play a game of Fibonacci solitaire. Keep track of your partial results like in the example above. [1, 5, 3, 6, 8, 7, 2, 4]

1
(1,5)
(1,5) 3
(1,5)(3,6)
(1,5)(3,6) 8
(1,5)(3,6) 7,8
(1,5) 2 (3,6) 7,8
(1,5)(2,4)(3,6) 7,8

4. Find the depth of the lower card in each pair for all your partial results in the previous problem. If a card hasn't been drawn for a given partial result, then its depth is undefined. In the example above, $\delta(1)$ is undefined until the fourth step.

(1,5) ⇒ $\delta(1)=1$

(1,5)3 ⇒ $\delta(1)=2$

(1,5)(3,6) ⇒ $\delta(1)=3$, $\delta(3)=1$

(1,5)(3,6) 8 ⇒ $\delta(1)=4$, $\delta(3)=2$

(1,5)(3,6) 7,8 ⇒ $\delta(1)=5$, $\delta(3)=3$

(1,5) 2(3,6) 7,8 ⇒ $\delta(1)=6$, $\delta(3)=3$

(1,5)(2,4)(3,6) 7,8 ⇒ $\delta(1)=7$, $\delta(2)=5$, $\delta(3)=3$

Rigging

What we really need to keep track of though is the depth of the lower cards in each pair when that pair is first created. In the example above, $\delta(4) = 1$ when the pair (4,6) was first made. Also $\delta(1) = 4$ when the pair (1,2) was first made. We will write this important information as a subscript, so that our new final result looks like $(1_4, 2), 3, (4_1, 6), 5$

$(1,5) \Rightarrow \delta(1)=1$
$(1,5)(3,6) \Rightarrow \delta(3)=1$
$(1,5)(2,4)(3,6)\,7,8 \Rightarrow \delta(2)=5$

5. Using your game from problem 3, determine the depth of the low card in each pair when that pair was first made. Write that information as a subscript on your final result.

$(1_1, 5)(2_5, 4)(3_1, 6)\ 7, 8$

8
6, 8
(6, 7) 8
3 (6, 7) 8
(3,4) (6, 7) 8
1 (3,4) (6, 7) 8
(1,2) (3,4) (6, 7) 8
(1,2) (3,4) 5 (6, 7) 8

6. Shuffle your deck and play Fibonacci solitaire. Write down all the partial results as well as the starting permutation, and find the depth of the lower card from each pair when that pair was first created. [8, 6, 7, 3, 4, 1, 2, 5]

$(1_6, 2)(3_4, 4)\ 5\ (6_2, 7)\ 8$

This information that you are writing as a subscript on your final results we will call the *rigging*. The rigging will allow us to play the game backward.

7. Ask your instructor for a backward playing instruction sheet. Did you get one?

YES

8. Follow the directions carefully and, starting from your final result in problem 6, play backward. Don't peek at your partial results from 6, just follow the directions and when you are done, check to see that you ended at your starting permutation from problem 6. If you ended elsewhere, make sure your rigging is correct and try again. Follow the instructions carefully.

9. Use the rigged pairing

$$1, (2_4, 6), (3_2, 10), 4, (5_3, 7), 8, 9$$

and play backward. Write the order of the deck you obtained as a result.

[9, 3, 10, 2, 6, 8, 5, 7, 4, 1]

10. Remember what we said last time about inverse functions and one-to-one functions. Because we can now undo the Fibonacci solitaire function, what does that tell you?

11. What is the domain of the Fibonacci solitaire function?

PERMUTATIONS

- OR -

ORDERS OF A DECK OF CARDS.

12. What is the new range of the Fibonacci solitaire function?

PAIRINGS of THE CARDS <u>WITH</u> RIGGING

15 Lesson 8: Winning Fibonacci solitaire

Understanding bijective correspondences need not wait until college algebra. It is clear to see that if a correspondence can be made from the objects of one set to the objects of another, then those sets must be of the same size. This lesson helps students to understand that, and helps them to see distinctly what sets are in bijective correspondence due to the Fibonacci solitaire algorithm. Students then work to count the number of objects in the domain and in the range. Students will determine how to count the number of rigged perfect pairings using their knowledge gained from lesson 3. These numbers are sufficient to calculate the probability of winning, and students will do so for a few values of n. The mathematics here is deep, but because the example is concrete, we hope that students will be able to more readily generalize to the abstract.

Notes to the instructor

Materials *Cards:* Students will need a deck of numbered cards. A deck of size 10 is definitely sufficient. Many of the subsequent lessons require only six cards. A deck of standard playing cards, partitioned into suits, with King, Queen, and Jack removed will give 4 sets of 10 cards when the Ace is considered as a 1. If playing cards, or simply numbered cards are unavailable, you can have students make their own cards using index cards or slips of paper.

Lesson worksheets: The activity below can be photocopied and should be distributed to students. An overhead can be made from the activity as well, if so desired, then students will work with their own paper to answer the questions projected.

Goals Students will use their intuitive understanding of correspondences to conjecture about numbers of objects in different sets.

Students will use previous knowledge of counting rules to count the number of rigged perfect pairings.

Students will use their conjecture about bijective correspondences to understand that the number of rigged perfect pairings is the same as the number of winning permutations.

Students will calculate the probability of winning using a deck of given size.

Prerequisite student knowledge Students should understand the definition of function.

Students should be able to reason about sets of objects.

Students should complete activity 3, or be familiar with combinatorial counting rules.

Students should be familiar with simple probability.

Possible student misconceptions The counting problems in this section are non-trivial. Students may develop a variety of counting arguments. There are many correct methods of counting the set of rigged perfect pairings. Unfortunately, there are even more incorrect ways to count this set. Students may convince themselves that they have found a correct way to count the set when they have either under-counted or over-counted. In each case it is the teachers' place to give an example of the over/under count so that students will continue to work.

Activity 8 Fibonacci Solitaire

May 2010 Cliff Smith

Name: ____

One-to-one correspondences

1. Suppose that there are two parking lots. One lot has trucks in it, and the other has cars. Suppose that every truck is a different color than every other truck. Also suppose that for every truck of a given color you can find exactly one car of that same color in the other lot. For example, if you see an orange truck, that means that there is exactly one orange car. What can you say about the number of trucks and the number of cars?

2. Imagine a room with a bunch of your friends in it. Some people are on the left side of the room, and some on the right. Every person on the left side of the room is pointing at someone on the right side. That person on the right side is pointing straight back at the person who is pointing at him. Suppose everybody is pointing at somebody. What can you say about the number of people on the left side of the room and the number of people on the right side of the room.

The previous two problems are examples of a *one-to-one correspondence.*

3. Count the number of chairs in your classroom *out loud.* How many are there?

What you've just done is create a one-to-one correspondence between the numbers 1, 2, 3, 4,... and the chairs in your room.

4. In your own words (two sentences minimum) write what it means for two sets of objects to be in a one-to-one correspondence.

5. What does a one-to-one correspondence tell us about the numbers of objects in these sets?

A function and its inverse can be thought of as a room full of people pointing at each other like in problem 2. The permutation [4, 6, 5, 1, 2, 3] and the rigged result $(1_4, 2), 3, (4_1, 6), 5$ would be two people on opposite sides of the room, both pointing at each other. This means that a function that has an inverse creates a one-to-one correspondence between its domain and its range.

6. What can we say about the sizes of the set of permutations of 6 objects, and the set of rigged results of Fibonacci solitaire played with 6 cards?

Counting winning games

What we're really interested in is the probability of winning a game of Fibonacci solitaire. It's hard to determine how many permutations end in winning games. However, we can handily count the winning rigged results. Since they each point at a specific permutation, meaning that they are in a one-to-one correspondence with winning permutations, they are the same number.

7. Work with a friend to find the number of winning final results when playing with 8 cards. Remember what you learned when counting permutations.

8. Given a winning game, how many different ways can you pick a rigging? Remember, riggings must be positive and less than or equal to the card's final depth. (how many different values can a given card have?)

9. How many winning, rigged results are there when playing with 8 cards?

10. How many permutations are there for an 8 card deck?

11. The probability of winning is given by

$$\frac{\# \ of \ winning \ permutations}{\# \ of \ permutations}.$$

What is probability of winning with an 8 card deck?

Activity 8 Fibonacci Solitaire

May 2010 Cliff Smith

Name: SOLUTIONS

One-to-one Correspondences

1. Suppose that there are two parking lots. One lot has trucks in it, and the other has cars. Suppose that every truck is a different color than every other truck. Also suppose that for every truck of a given color you can find exactly one car of that same color in the other lot. For example, if you see an orange truck, that means that there is exactly one orange car. What can you say about the number of trucks and the number of cars?

SINCE, FOR EVERY TRUCK, THERE'S ONE CAR OF THE SAME COLOR, THEN THE NUMBER OF TRUCKS IS THE SAME AS THE NUMBER OF CARS.

2. Imagine a room with a bunch of your friends in it. Some people are on the left side of the room, and some on the right. Every person on the left side of the room is pointing at someone on the right side. That person on the right side is pointing straight back at the person who is pointing at him. Suppose everybody is pointing at somebody. What can you say about the number of people on the left side of the room and the number of people on the right side of the room.

SO, SAY, KEVIN IS POINTING AT KAREN, THEN KAREN IS POINTING AT KEVIN. SO THEY AREN'T POINTING AT ANYBODY ELSE. FOR EVERY PERSON ON THE LEFT SIDE, THERE IS ONE PERSON ON THE RIGHT.

THE NUMBER OF PEOPLE ON THE LEFT IS THE SAME AS THE NUMBER OF PEOPLE ON THE RIGHT.

The previous two problems are examples of a *one-to-one correspondence.*

3. Count the number of chairs in your classroom *out loud.* How many are there? 29

What you've just done is create a one-to-one correspondence between the numbers 1, 2, 3, 4,... and the chairs in your room.

4. In your own words (two sentences minimum) write what it means for two sets of objects to be in a one-to-one correspondence.

FOR EVERY OBJECT IN ONE OF THE SETS, THERE IS EXACTLY ONE CORRESPONDING OBJECT IN THE OTHER SET.

5. What does a one-to-one correspondence tell us about the numbers of objects in these sets? THE NUMBER OF OBJECTS IN ONE SET IS THE SAME AS THE NUMBER OF OBJECTS IN THE OTHER.

A function and its inverse can be thought of as a room full of people pointing at each other like in problem 2. The permutation [4, 6, 5, 1, 2, 3] and the rigged result $(1_4, 2), 3, (4_1, 6), 5$ would be two people on opposite sides of the room, both pointing at each other. This means that a function that has an inverse creates a one-to-one correspondence between its domain and its range.

6. What can we say about the sizes of the set of permutations of 6 objects, and the set of rigged results of Fibonacci solitaire played with 6 cards?

THE NUMBER OF PERMUTATIONS OF 6 THINGS,

AND THE NUMBER OF RIGGED PAIRINGS ARE THE SAME

Counting Winning Games

What we're really interested in is the probability of winning a game of Fibonacci solitaire. It's hard to determine how many permutations end in winning games. However, we can handily count the winning rigged results. Since they each point at a specific permutation, meaning that they are in a one-to-one correspondence with winning permutations, they are the same number.

7. Work with a friend to find the number of winning final results when playing with 8 cards. Remember what you learned when counting permutations. $7 \cdot 5 \cdot 3 \cdot 1 = 105$

8. Given a winning game, how many different ways can you pick a rigging? Remember, riggings must be positive and less than or equal to the card's final depth. (how many different values can a given card have?)

$7 \cdot 5 \cdot 3 \cdot 1 = 105$

9. How many winning, rigged results are there when playing with 8 cards?

$105 \cdot 105 = 11025$

10. How many permutations are there for an 8 card deck?

$8! = 40320$

11. The probability of winning is given by

$$\frac{\#\ of\ winning\ permutations}{\#\ of\ permutations}.$$

What is probability of winning with an 8 card deck?

$$\frac{11025}{40320} \approx .273$$

16 Reflection on classroom implementation

In an attempt to estimate the effectiveness of such activities in the classroom, I implemented two previous versions of the activities outlined above in two separate classroom experiments. One of these was with a mixed group of undergraduate and graduate students who were taking a discrete mathematics course designed for current and future mathematics teachers. The second of these experiments took place in an intermediate algebra course at a nearby community college. In the first class, the focal points of the lesson were the counting process and the use of bijective functions to simplify object counting. For the second class, I implemented a lesson designed to emphasize the differences between injective and non-injective functions in the hopes that such a lesson might help students to understand the abstract definitions given in their textbooks. Changes have been made to these lessons for several reasons, but the goals addressed in my classrooms are still addressed by the modified lessons in this article.

While designing the lessons, I was consistently questioning whether a certain question or task might reduce the cognitive demand of the activity. There was a constant struggle to balance mathematical demand while guiding student attention. In their book [6], Boaler and Humphreys develop a method of determining the cognitive demand of classroom activities. If an activity gives too much information, or if questions divide the activity into steps that are simply procedural, much of the cognitive demand of the task can be removed. However, if not enough support is given to the students they may spend much of their time following a path that will not lead to success, or they may stop trying and degenerate into non-mathematical activity (2005). Understanding this, I constantly asked myself if a certain question would remove the student's opportunity to think deeply, while simultaneously asking whether the questions I was asking provided an appropriate amount of scaffolding. I believe that the lessons I created were appropriately demanding of the student, however that demand may have been misplaced. In an attempt to build activities that were inquiry oriented, I required students to develop the reverse algorithm themselves. At times the majority of the cognitive demand on the student seemed to be that of creating the algorithm. I want the demand to be in developing arguments that the algorithm is injective, or that the inverse algorithm is an inverse function. Students should be occupied in learning mathematics that is applicable to other situations, not simply this algorithm. They seemed to enjoy the puzzle of trying to construct the reversal algorithm. Typically, students were able to see that smaller singletons need be removed first, and that a pair's top needs to be removed before the bottom. However, this logical puzzle does not necessarily lead to mathematical activity. The lessons below have been modified to place more of the demand on the counting arguments and functional analysis, while removing much of the demand in the section about the algorithm. It is not necessary that students construct the reverse algorithm, but that they understand how to use it, and that it works.

In both cases, student interest was high. Students in each of the experimental classrooms indicated that they liked the activities. There is some question as to whether the physical manifestation of the function in the form of an algorithm on playing cards allowed greater understanding of the underlying function. I did not attempt to teach the algorithm without cards, but this could be an interesting question for further study. The lessons I hoped to implement were slightly ambitious in the sense that I underestimated the time that would be required for students to be able to complete the tasks. Modifications were made to the lessons to correct this mistake. These modifications range from splitting a lesson into several, to giving more direct instruction in the use of the algorithms, allowing students more time to deal with the mathematics.

In preparing the lesson for the first class, I wanted to reduce some of the confusion and direct instruction necessary to teach the standard notation in which we write pairings that allows determination of the rigging. It is difficult to explain this notation in writing, and often is difficult even with examples. It often takes several examples to ensure that all students understand the specifics. I believed that I could modify the algorithm in such a way that the cards on students' tables would be ordered in a way that directly translated to the standard written form. I did so quickly and perhaps over-confidently. When reading the algorithm, I was not sufficiently objective. I believed it would act as I hoped, but I was sorely incorrect. Here is the algorithm that I presented to the students.

Wrong 1: If the drawn card is smaller in value than the smallest on the table, place it face up to the left of the cards on the table.

Wrong 2: If the card is larger than any on the table, slide it under the smallest unpaired card on the table. These two cards are now a pair and will not be separated for the rest of the game.

Close inspection shows that this is not equivalent to the original algorithm. This led to great confusion in the classroom about what the algorithm was supposed to do and how to play the game. In a class of undergraduate or high school students, this may have derailed the entire activity. At the very least, it would have necessitated a large amount of time in which the correct algorithm was written on the board and students learned to play. Additionally, If a person who was not familiar with the algorithm tried to use the lesson, they may not have been able to produce the correct algorithm. This mistake is a clear reminder that any modifications to a lesson require a significantly objective eye to determine whether the changes even make sense. Since then, I have created the algorithm that does what I wanted the bad algorithm above to do, but it is unwieldy and unnecessarily complicated.

Revision 1: If the drawn card is smaller in value than the smallest unpaired card on the table or if there is no unpaired card on the table, place it on the table in numerical order in relation to the single cards and the smaller cards of each pair.

Revision 2: If the card is larger than the smallest unpaired card on the table, slide it under that smallest unpaired card. These are now a pair and will stay together throughout the rest of the game.

These instructions are difficult to follow, and clarification would require the addition of further vocabulary, or perhaps breaking the algorithm into three cases instead of just two. Further, I believe that these more complicated instructions would make reverse play more difficult for students to understand. Whereas, with the original algorithm, it is clear that the larger card in every pair comes later in the permutation, this more complicated algorithm seems to obscure that information by requiring the constant moving and reorganizing of cards during game play. Retaining the original, simpler algorithm seems the best choice, and the small amount of time required to ensure that students understand the standard notation is a necessary step. A step that, I believe, is best accomplished through direct instruction.

References

[1] Al-Agha K.J., & Greechie R.J. (2008). *Order dimension of orthomodular amalgamations over trees.* Order. 25 (1), 55-68.

[2] Asiala, M., Kleiman, J., Brown, A., & Mathews, D. (1998). *The Development of Students' Understanding of Permutations and Symmetries.* International Journal of Computers for Mathematical Learning. 3 (1), 13.

[3] Asiala, M., Dubinsky, E., Mathews, D. M., Morics, S., & Okra, A. (1997). *Development of Students' Understanding of Cosets, Normality, and Quotient Groups.* Journal of Mathematical Behavior. 16 (3), 241-309.

[4] Bates, C., Bundy, D., Perkins, S., & Rowley, P. (2004). *Commuting Involution Graphs in Special Linear Groups.* Communications in Algebra. 32 (11), 4179-4196.

[5] Bax, E. T. (1994). *Algorithms to count paths and cycles.* Information Processing Letters. 52 (5), 249.

[6] Boaler, J., & Humphreys, C. (2005). *Connecting mathematical ideas: Middle school video cases to support teaching and learning.* Portsmouth, NH: Heinemann.

[7] Devillers A., & Giudici M. (2008). *Involution graphs where the product of two adjacent vertices has order three.* Journal of the Australian Mathematical Society. 85 (3), 305-322.

[8] Gnedin, A., & Kerov, S. (2000). *The Plancherel measure of the Young-Fibonacci graph.* Mathematical Proceedings of the Cambridge Philosophical Society. 129 (3), 433-446.

[9] Gnedin, A., & Kerov, S. (2002). *Fibonacci Solitaire. Random Structures and Algorithms.* 20, 71-88.

[10] Gnedin, A., & Kerov, S. (2005). *Derangement Characters of the Finite General Linear Group.* Algebras and Representation Theory. 8 (2), 255-274.

[11] National Governors Association and Council of Chief State School Officers. (2009). *College and Career Readiness Standards for Mathematics (Draft for Review and Comment).* Common Core Standards Initiative, http://www.corestandards.org/Files/MathStandardsSources.pdf.

[12] National Governors Association and Council of Chief State School Officers. (2009). *Common Core State Standards for Mathematics (draft).* Common Core Standards Initiative, http://www.corestandards.org/Files/K12MathStandards.pdf.

[13] Perkins, S. (2006). *Commuting involution graphs for $\tilde{A}_n$.* Archiv Der Mathematik. 86 (1), 16-25.

[14] Stein, M. K. (2000). *Implementing standards-based mathematics instruction: A casebook for professional development*. Ways of knowing in science series. New York: Teachers College Press.

[15] Weisstein, Eric W. "Gamma Function." From *MathWorld*–A Wolfram Web Resource. http://mathworld.wolfram.com/GammaFunction.html