THE REVERSED ZECKENDORF GAME

ZOË X. BATTERMAN, ADITYA JAMBHALE, STEVEN J. MILLER, AKASH L. NARAYANAN, KISHAN SHARMA, ANDREW K. YANG, AND CHRIS YAO

Abstract. Zeckendorf [Ze] proved that every natural number n can be expressed uniquely as a sum of non-consecutive Fibonacci numbers, called its *Zeckendorf decomposition.* Baird-Smith, Epstein, Flint, and Miller [BEFM1] created the *Zeckendorf game*, a two-player game played on partitions of n into Fibonacci numbers which always terminates at a Zeckendorf decomposition, and proved that Player 2 has a winning strategy for $n \geq 3$. Since their proof was non-constructive, other authors have studied the game to find a constructive winning strategy, and lacking success there turned to related problems. For example, Cheigh, Moura, Jeong, Duke, Milgrim, Miller, and Ngamlamai [CMJDMMN] studied minimum and maximum game lengths and randomly played games. We explore a new direction and introduce the *reversed Zeckendorf game*, which starts at the ending state of the Zeckendorf game and flips all the moves, so the reversed game ends with all pieces in the first bin. We show that Player 1 has a winning strategy for $n = F_{i+1} + F_{i-2}$ and solve various modified games.

1. Introduction and Main Results

1.1. **History.** The Fibonacci numbers, which for uniqueness results in decompositions requires us to define them by $F_1 = 1$, $F_2 = 2$, and $F_{n+1} = F_n + F_{n-1}$, is a sequence with many interesting properties which have been widely studied. With this choice of indexing, Zeckendorf [Ze] proved that every natural number n has a unique *Zeckendorf decomposition*, which expresses n as the sum of distinct, non-adjacent Fibonacci numbers. Note the decomposition would no longer be unique if we considered there to be two ones or a zero in the Fibonacci sequence. An example of such a Zeckendorf decomposition is

$$2024 \;=\; 1597 + 377 + 34 + 13 + 3. \tag{1.1}$$

Building upon this, Baird-Smith, Epstein, Flint and Miller in [BEFM1] and [BEFM2] created the *Zeckendorf game.* The Zeckendorf game, which we will refer to as the *forwards game*, starts with a natural number n. A game state consists of a partition of n into Fibonacci numbers, constituents of which we call *chips.* We say the collection of F_i's in this partition is the i^{th} bin, with its cardinality h_i called the *height* of the bin. There are two types of moves.

(i) *Combine*: If $h_i > 0$ and $h_{i-1} > 0$, then the move is

$$\begin{aligned} F_{i-1} + F_i \;&\mapsto\; F_{i+1}, \\ 2F_1 \;&\mapsto\; F_2. \end{aligned} \tag{1.2}$$

In other words, we remove one chip from each of the i^{th} and $(i-1)^{\text{th}}$ bins and add a chip to the $(i+1)^{\text{th}}$ bin.

This work was completed as part of the 2023 SMALL REU program at Williams College. It was supported in part by NSF Grants DMS1561945 and DMS1659037, Williams College, and Churchill College, Cambridge.

(ii) *Split*: If $h_i \geq 2$ with $i > 2$, then we have the move

$$\begin{aligned} 2F_i &\mapsto F_{i-2} + F_{i+1}, \\ 2F_2 &\mapsto F_3 + F_1. \end{aligned} \tag{1.3}$$

Note that each of these moves keeps the total sum of the values of the chips constant at n. The forwards game starts with n ones and proceeds until a player can't make a move, in which case the player who can't move loses. This makes the Zeckendorf game a normal, impartial, combinatorial game.[1]

The authors in [BEFM1] showed the game for n always terminates at n's Zeckendorf decomposition, and they showed, nonconstructively, Player 2 wins for all $n \geq 3$. After this result was published, many authors have investigated the properties of this game and various modifications of it such as possible game lengths, random games, etc; see [CDHK+6i], [CDHK+6ii], [BCDD+15], [BEFM2], [CMJDMMN], [GMRVY], [LLMMSXZ], and [MSY].

We introduce another version of this game, the *reversed Zeckendorf game*, where the players play with all the moves reversed and the starting and ending positions exchanged; for the full definition, see §3. This game demonstrates a more complex winning structure than the forwards game.[2]

1.2. **Main Results.** First, in §3, we analyze the reversed Zeckendorf game, look at its winning structure, and make further conjectures. As in the literature, we denote the golden ratio $\frac{1+\sqrt{5}}{2}$ by ϕ. Moveover, for brevity in proofs, we will say that a player "has a win" (respectively "has a loss") by which we mean "has a forced winning strategy" (respectively "the opponent has a force winning strategy"). We state our main results.

Theorem 1.1. *Player 1 has a winning strategy for the reversed Zeckendorf game when*

$$n = F_{i+1} + F_{i-2}. \tag{1.4}$$

We have both a nonconstructive and constructive proof for this theorem. The nonconstructive proof involves a strategy-stealing argument, and the constructive proof provides an explicit strategy.

Conjecture 1.2. *For the reversed Zeckendorf game, Player 2 has a winning strategy for infinitely many n.*

Conjecture 1.3. *In the limit, the percent of the time Player 1 has a winning strategy for the reversed Zeckendorf game is $\phi^{-1} \approx .618$.*

Continuing in §3.1, we transfer results from [CMJDMMN] to get results about randomly played reversed Zeckendorf games as well as bounds on how long the game can last. These results are summarized here.

Theorem 1.4. *Let $Z(n)$ be the number of terms in the Zeckendorf decomposition of n.*

(i) The shortest possible reversed Zeckendorf game is $n - Z(n)$.

(ii) An upper bound on the longest possible reversed Zeckendorf game is

$$\lfloor \phi^2 n - Z_I(n) - 2Z(n) + \phi - 1 \rfloor, \tag{1.5}$$

where $Z_I(n)$ is the sum across the indices of the Fibonacci numbers in the Zeckendorf decomposition.

[1] The terms normal and impartial are defined in §2.

[2] Here, "winning structure" refers to when each player has a forced win.

Theorem 1.5. *For any integer $Z \geq 1$ and $z \in \{0, 1, \dots, Z-1\}$, we have*

$$\lim_{N\to\infty} \mu_N(\textit{game length equals } z \textit{ mod } Z) \;=\; \lim_{N\to\infty} \mathbb{P}_N(\textit{game length equals } z \textit{ mod } Z) \;=\; \frac{1}{Z}, \tag{1.6}$$

where μ_N and $\mathbb{P}_N$ are two different probability measures, defined in §3.1, being placed on the space of all possible games starting at the Zeckendorf decomposition of N.

In §4, we analyze winning strategies for different starting positions of the reversed Zeckendorf game. That is, we assume the starting position is not at the Zeckendorf decomposition but at some other partition of n into Fibonacci numbers. Specifically, we completely solve the game when the starting position only involves ones, twos, and threes.

Theorem 1.6. *For any game starting with a ones, b twos, and c threes, we have the following forced wins.*

a	b	c		*Player having forced win*
Even	*Even*	*Even*		*Player 2*
Odd	*Odd*	*Odd*		*Player 1*
Even	*Odd*	*Even*		*Player 1*
Odd	*Even*	*Odd*		*Player 1*
Odd	*Even*	*Even*	$a > c$	*Player 2*
Odd	*Even*	*Even*	$a < c$	*Player 1*
Even	*Even*	*Odd*	$a > c$	*Player 1*
Even	*Even*	*Odd*	$a < c$	*Player 2*
Even	*Odd*	*Odd*		*Player 1*
Odd	*Odd*	*Even*		*Player 1*

Finally, in §5, we conclude with a further modification of this game, which we call the *build up game*, and we completely solve it. The proof involves Nim-like strategies—strategies that involve forcing the opponent along game states associated to a fixed residue class modulo some integer.

2. Reversed Games

Intuitively, two players playing a reversed game should look, if time is running backwards, like the forwards game. As such, we want a move in the reversed game from a state A to a state B to be legal if and only if moving from B to A is legal in the forwards game. To formalize this, we add the assumption that the game is *normal and impartial.*

Definition 2.1. *We say a game is* normal and impartial *when*

(i) the allowable moves depend only on the position and not on player order, and
(ii) the last player to move wins.

We can consider a normal, impartial game as a game played on a directed graph. Players take turns moving a game piece across the graph until a terminal node is reached (e.g., a node where there are no moves).

Definition 2.2 (**Reversed Game**). *Let G be the associated directed graph of a normal, impartial game. Suppose that the graph has a unique loss node and unique starting node. Then the* reverse *of that game is obtained by reversing all the edges in G and playing from the loss node to the starting node. A player loses if they run out of moves.*

Remark 2.3. *Under this definition, the reverse of a game is also normal and impartial.*

Example 2.4. *Consider the reversed game for standard Nim, where players take turns removing* 1*,* 2*, or* 3 *from an natural number* $n \equiv 1 \pmod 4$*, never going below* 1*, with the last player to move winning. Note standard Nim is a normal, impartial game with starting node* n *and ending node* 1*.*

The reversed game would be played by starting at 1 *and adding either* $1, 2,$ *or* 3 *(never going above* n*), with the first person to make the number* n *winning.*

This is transparently the same as Nim, with the same winning strategy for Player 2*. Thus, this game does not demonstrate any new behavior when reversed: The winning structure is still the same as Player* 2 *always wins with the same strategy.*

In general, reversed games do not exhibit the same winning structure or winning strategies. As we will see, reversed Chomp has a similar winning structure but different winning strategy, and the reversed Zeckendorf game has an entirely different winning structure.

2.1. **Reversed Chomp.**

As a useful example of a reversed game, we solve the reversed Chomp game. First, we introduce the game of *Chomp*. Chomp is a normal, impartial game played on a rectangular board with N rows and M columns. Players take turns choosing a square on the board and "eating" it, removing that square and all the squares both above and to the right of it. The player forced to eat the "poisoned square" in the bottom left corner loses the game. We consider the game played with two players and ignore the trivial game $N = M = 1$.

We can consider this game as played on a directed graph with a starting node representing the full board and an ending node representing the poisoned square. It can be shown by a "strategy-stealing argument" that for all N, M, Chomp is a win for Player 1. However, constructing an explicit winning strategy proves difficult.

Consider *reversed Chomp*, which starts from the poisoned square and ends with the full board. Without loss of generality, assume $N \geq M$. When $M = 1$, Player 1 wins in one move.

We provide a constructive proof that $M > 1$ is always a win for Player 2 by giving the following winning strategy.

(i) **Case** $M = 2$: Player 1 has three options for their first move: complete the bottom row, complete the leftmost column, or partially complete the leftmost column. The first two choices allow Player 2 to win in one move.

Suppose Player 1 partially completed the leftmost column to a height of $1 < h < N$ squares. Then Player 2 can respond by filling in the other column to a height of $h - 1$. The game has now been reduced to another game with 2 columns but with fewer rows than we started with (see figure 1).

Player 1 is presented with the same three choices. Player 2 can continue responding in the above manner until we reach the game with 2 rows and 2 columns. From this position, regardless of Player 1's choice, Player 2 wins in one move.

(ii) **Case** $M > 2$: Player 1 has four options for their first move: complete the bottom row, complete the leftmost column, partially complete the bottom row, or partially complete the leftmost column. The first two choices allow Player 2 to win in one move.

If Player 1 partially completes the bottom row to a length of $1 < l < M$ squares, then Player 2 can respond by filling in the leftmost $l - 1$ columns (see figure 2). This reduces to a game with the same number of rows and fewer columns than we started with. Similarly, if Player 1 partially completes the leftmost column to a height of $1 < h < N$ squares, then Player 2 can respond by filling in the bottom $h - 1$ rows

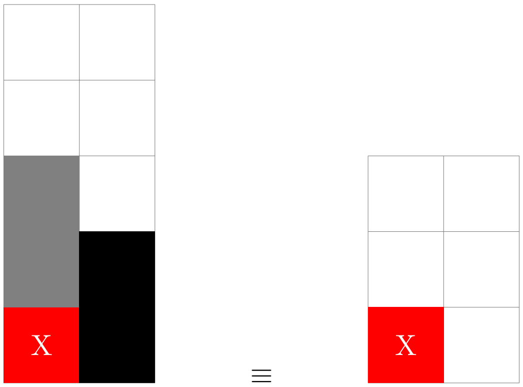

Figure 1. $M = 2$. Gray indicates move by Player 1, black indicates move by Player 2

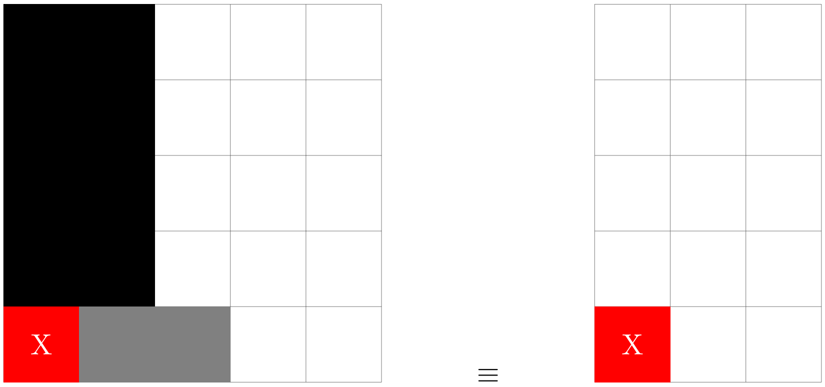

Figure 2. $M > 2$, Player 1 partially completes bottom row.

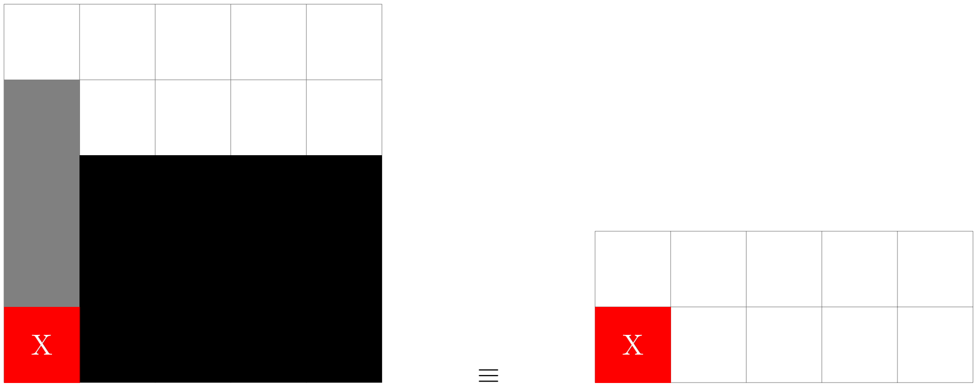

Figure 3. $M > 2$, Player 1 partially completes leftmost column.

which reduces to a game with the same number of columns but fewer rows (see Figure 3).

Either way, Player 1 is presented with the same four choices. Player 2 can continue responding in this manner until the game is reduced to a state with either only 2 rows or 2 columns. By the previous case, this is a win for Player 2.

Remark 2.5. *Along with the winning strategy, the winning structure for reversed Chomp is also different than normal Chomp. Player 1 wins reversed Chomp for all N when $M = 1$. This is an infinite number of games but not a positive proportion. In forward Chomp, Player 2 only wins a finite number of games, namely just the trivial game.*

3. The Reversed Zeckendorf Game

Following our definition of the reversed game in Definition 2, we give an explicit construction for the *reversed Zeckendorf game.*

The terminology is the same as in the forwards game (with the starting and ending positions flipped, so we always start at the Zeckendorf decomposition of n) except we rename each move.

(i) *Split*: We remove one chip from the $(i+1)^{\text{th}}$ bin and place one chip each in the i^{th} and $(i-1)^{\text{th}}$ bins.

$$\begin{aligned} F_{i+1} &\mapsto F_{i-1} + F_i, \\ F_2 &\mapsto 2F_1. \end{aligned} \tag{3.1}$$

(ii) *Combine*: For $i > 2$, we have

$$\begin{aligned} F_{i-2} + F_{i+1} &\mapsto 2F_i, \\ F_3 + F_1 &\mapsto 2F_2. \end{aligned} \tag{3.2}$$

Theorem 1.1. *Player 1 has a winning strategy for the reversed Zeckendorf game when*

$$n = F_{i+1} + F_{i-2}. \tag{1.4}$$

We present two proofs, one constructive and the other nonconstructive.

Proof. (Nonconstructive) By way of contradiction, suppose Player 2 has a winning strategy. If Player 1 chose to combine for their first move, Player 2 has a forced win starting at the state $2F_i$. There is only one move in this position, which means Player 2 has a forced win with Player 1 starting at the state $F_i + F_{i-1} + F_{i-2}$.

However, Player 1 could choose to instead split the $(i+1)^{\text{th}}$ bin, which makes it Player 2's turn at the state $F_i + F_{i-1} + F_{i-2}$. Now, Player 1 can steal the strategy from Player 2 to have a forced win in the starting state, a contradiction. □

Remark 3.1. *The constructive proof relies on Lemma 4.1. As a corollary of this lemma, we have a constructive proof for why the game state $2F_i$ is a win for whoever goes second. Before that, Player 1 starts with combining to bring the game into this situation.*

The next natural question is whether or not the same can be said of Player 2.

Conjecture 1.2. *For the reversed Zeckendorf game, Player 2 has a winning strategy for infinitely many n.*

To investigate this conjecture, we directly computed which player had a forced winning strategy for $n \leq 129$ (see Appendix A.1). The code for computing which player has a forced winning strategy is listed in Appendix B and has a algorithmic complexity of $O(\exp(\sqrt{n})$. For $n = 129$, the program took close to 2 hours.

Figure 4 shows the proportion Player 2 wins. We plot n versus the percent of games won by Player 2 in games 2 through n.

For the first $n \leq 129$, the average number of Player 1 wins is $80/129 \approx .620$. The average number of wins seems to stabilize computationally. Combining these two together, we are led to the following conjecture.

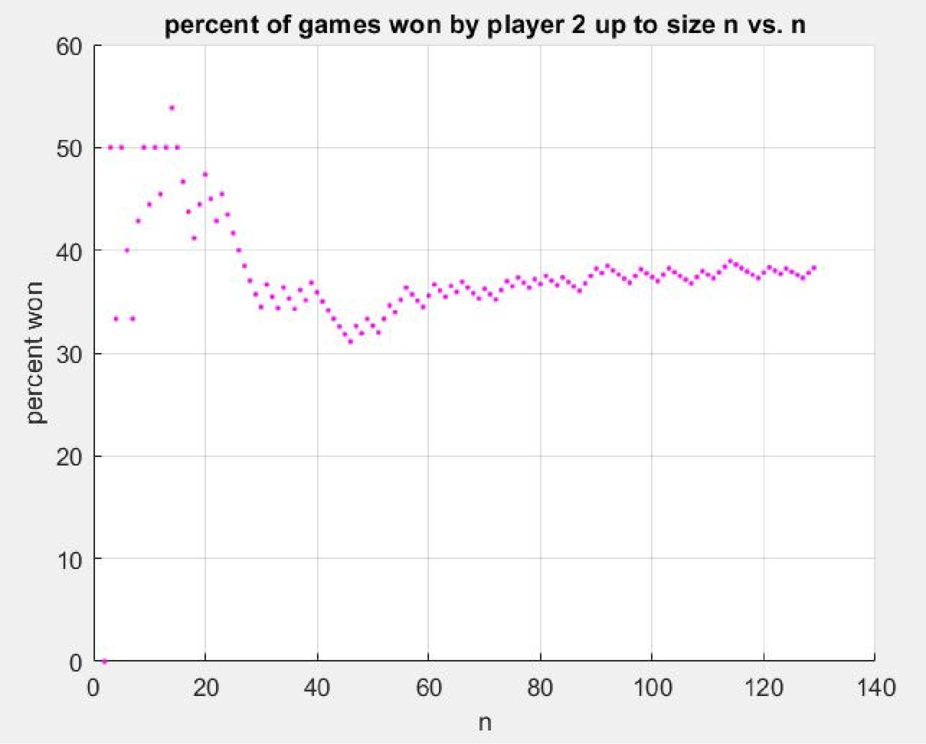

FIGURE 4. Percent of games $\leq n$ won by Player 2

Conjecture 1.3. *In the limit, the percent of the time Player 1 has a winning strategy for the reversed Zeckendorf game is* $\phi^{-1} \approx .618$.

This is a natural conjecture due to the connection between the Fibonacci numbers and the golden ratio. Work on determining the winner is provided in §4.

3.1. **Other Facts.** We gather various results about the reversed Zeckendorf game which follow from already-known facts about the forwards game. Since we have merely reversed the arrows in the forward game tree to produce the reversed game, many properties about the forwards game extend immediately as corollaries. One property that extends is the lengths of games. From [BEFM1] for the lower bound and [CMJDMMN] for the upper bound, we have the following.

Theorem 1.4. *Let $Z(n)$ be the number of terms in the Zeckendorf decomposition of n.*

(i) The shortest possible reversed Zeckendorf game is $n - Z(n)$.

(ii) An upper bound on the longest possible reversed Zeckendorf game is

$$\lfloor \phi^2 n - Z_I(n) - 2Z(n) + \phi - 1 \rfloor, \tag{1.5}$$

where $Z_I(n)$ is the sum across the indices of the Fibonacci numbers in the Zeckendorf decomposition.

Theorems 1.8 and 1.9 from the paper [CMJDMMN] also transfer.

Finally, we have results on randomly played games with one subtlety. The paper discusses two different probability measures put on the space of all possible games with a fixed n. One of them simply assigns the same probability to each game (denoted μ_n), as well as assigning each game according to the probability of playing it if each player picked a random move each turn, uniformly from the possible moves (denoted $\mathbb{P}_n$).

Now, the first measure μ assigns the same probability to each game as its corresponding flipped game in the reversed version. As such, statements about this probability measure easily translate as a corollary. However, the measure $\mathbb{P}_n$ requires more work than deducing the result for the reversed game from the original game. In this case, modifications and checks are required throughout Section 4 of their paper until Lemma 4.5. In the end, the main result still holds.

Theorem 1.5. *For any integer $Z \geq 1$ and $z \in \{0, 1, \ldots, Z-1\}$, we have*

$$\lim_{N\to\infty} \mu_N(\text{game length equals } z \text{ mod } Z) \ = \ \lim_{N\to\infty} \mathbb{P}_N(\text{game length equals } z \text{ mod } Z) \ = \ \frac{1}{Z}, \tag{1.6}$$

where μ_N and $\mathbb{P}_N$ are two different probability measures, defined in §3.1, being placed on the space of all possible games starting at the Zeckendorf decomposition of N.

This theorem works for a more general setting than that of the two player game. The game can be modified to Z many players, wherein the players take turns in sequence and the last player who moves wins. The theorem can then be interpreted as saying that in the Z player game, where each player chooses their move uniformly at random from their possible choices, any given player will win approximately $1/Z$ percent of the time.

For the specific case of two players and an even n, we have the following.

Theorem 3.2. *For the Reversed Zeckendorf Game player on an even n,*

$$\mathbb{P}_n(\text{game length equals } 1 \text{ mod } 2) \ = \ \frac{1}{2}. \tag{3.3}$$

Proof. Equivalently, we can show that each player, playing randomly on each turn, each has a 1/2 probability of winning.

At some point, every game will have the $h_3 = 1$ and $h_i = 0$ for all $i > 3$. Furthermore, we can continue to the move that gets rid of the three. We have

$$n \ = \ h_1 + 2h_2 + 3. \tag{3.4}$$

Since n is even, this means $h_1 > 0$, so at the final turn $h_3 = 1$, the player has 2 choices: (i) combine a one and a three or (ii) split the three.

Once there are no threes (i.e., $h_3 = 0$), the game only has one option, namely splitting the two until there are no moves left. The winner is determined by the parity of the second bin. However, (i) will yield the height of bin 2 equals $h_2 + 2$ and (ii) will yield the height of bin 2 equals $h_2 + 1$.

There is an equal chance the player will lose or win, and nothing that came before or after matters. We conclude the probability either player wins is 1/2, as desired. □

4. Other Starting Positions

A natural modification of the reversed Zeckendorf game is to alter the starting position. Instead of starting at the Zeckendorf decomposition, we instead specify a starting position and start the game from there. In fact, such problems are often tractable as the starting and ending positions are well-understood. Contrast this with applying the same idea to the forwards game: In this case, we would still be headed towards the Zeckendorf decomposition, which can vary as n changes.

Lemma 4.1. *If the starting position of the reversed Zeckendorf game has all bins of even height, then Player 2 has a winning strategy.*

Proof. The proof uses a "copycat" strategy that is, if Player 1 does a move, Player 2 plays the exact same move. We must show the following:

(i) if Player 1 makes a move, then Player 2 can also make that move, and
(ii) after Player 2 moves, all the bins are of even height.

If these are both the case, then Player 1 will have to move on a position where there are only ones and twos. At this point, there will be an even number of twos, so the game from here on is deterministic; both players will split twos until there are no twos left. Since the parity of the twos is even, this means it will be Player 1's turn when there are no twos left, so they lose.

Consider Player 1 making a split move so that

$$\begin{aligned} h_i &\mapsto h_i - 1, \\ h_{i-1} &\mapsto h_{i-1} + 1, \text{ and} \\ h_{i-2} &\mapsto h_{i-2} + 1. \end{aligned}$$

Since h_i is even, then $h_i - 1$ is odd. Thus, $h_i - 1 \geq 1 > 0$. This means Player 2 can legally make the same splitting move with the net effect of

$$\begin{aligned} h_i &\mapsto h_i - 2, \\ h_{i-1} &\mapsto h_{i-1} + 2, \text{ and} \\ h_{i-2} &\mapsto h_{i-2} + 2, \end{aligned}$$

with all the other bins unchanged, so all the heights are still even. We can do a similar argument for the combine move which proves (1) and (2). □

We also solve the game when the starting position consists only of ones, twos, and threes. We denote the starting position with a ones, b twos, and c threes as the ordered triple (a, b, c) from this point on. We will also often use the notation that an O or E in the place of the element of the tuple means that the corresponding variable is odd or even respectively (i.e., (E, E, E) means that a, b and c are all even). Further, O' and E' mean that the corresponding variable is odd or even, but independent from the previous O and E.

Theorem 1.6. *For any game starting with a ones, b twos, and c threes, we have the following forced wins.*

a	*b*	*c*		*Player having forced win*
Even	*Even*	*Even*		*Player 2*
Odd	*Odd*	*Odd*		*Player 1*
Even	*Odd*	*Even*		*Player 1*
Odd	*Even*	*Odd*		*Player 1*
Odd	*Even*	*Even*	$a > c$	*Player 2*
Odd	*Even*	*Even*	$a < c$	*Player 1*
Even	*Even*	*Odd*	$a > c$	*Player 1*
Even	*Even*	*Odd*	$a < c$	*Player 2*
Even	*Odd*	*Odd*		*Player 1*
Odd	*Odd*	*Even*		*Player 1*

Proof. The first case is a corollary of Lemma 4.1, since all the bins have even height. The next three cases follow from the first case as Player 1 simply splits the 3, splits the 2, or combines the 1 and the 3 respectively to bring the game state into all three bins being even.

The next few cases are a bit more work. We show the (Odd, Odd, Even) case, where $a > c$, and put the rest in Appendix A.2.

For this case, we assume a is odd and b and c are even with $a > c$. We show Player 2 wins by force.

First, assume Player 1 splits one of the threes. This gives the game state

$$(a+1, b+1, c-1). \tag{4.1}$$

Since c is even, we have $c - 1 \geq 1$. This allows Player 2 to split another three, yielding

$$(a+2, b+2, c-2). \tag{4.2}$$

Note that we have $a + 2 > c - 2$. Moreover, $a - 2$ is odd while $b + 2$ and $c - 2$ remain even. By induction, Player 2 wins.

Next, assume Player 1 splits one of the twos. This yields the game state

$$(a+2, b-1, c). \tag{4.3}$$

Since b is even, we have $b - 1 \geq 1$. This allows Player 2 to split another two, yielding

$$(a+4, b-2, c). \tag{4.4}$$

Note we have $a + 4 > c$. Moreover, $a + 4$ is odd while $b - 2$ and c remain even. By induction, Player 2 wins.

Finally, assume Player 1 combines a one and a three. This yields the game state

$$(a-1, b+2, c-1). \tag{4.5}$$

Since c is even, we have $c - 1 \geq 1$. Moreover, since $a > c$ by assumption, we have $a > 1$. In particular, Player 2 is able to perform another combine, leading to

$$(a-2, b+4, c-2). \tag{4.6}$$

Note $a - 2 > c - 2$. Moreover, all parities are preserved. By induction, Player 2 wins. □

Remark 4.2. *We summarize two key points in the above proof.*

(i) The method in the above Theorem is constructive, so following it carefully provides a winning strategy.

(ii) The last two cases appear to not depend on whether $a > c$ or $a < c$, but the strategy to win changes in those cases.

5. The Build Up Game

We move on to the *Build Up 1-2-3 game.* In this game, two players, say Player 1 and Player 2, take turns putting down a 1, 2 or 3 until their sum equals exactly n, generating an ordered triple (a, b, c). After reaching exactly n, the players start playing the reverse Zeckendorf game on this ordered triple and start with the player who did not move last. Whoever wins this reversed Zeckendorf game wins the whole game. We solved this game.

Theorem 5.1. *For $n \neq 4$,*

$$n = 4 \implies \text{Player 1 wins}, \tag{5.1}$$

$$n \text{ odd} \implies \text{Player 1 wins}, \tag{5.2}$$

$$n \neq 4 \text{ even} \implies \text{Player 2 wins}. \tag{5.3}$$

Proof. Using Theorem 1.6, we know who wins in each ordered triple. We provide a constructive proof in which we split the game into cases mod 4. We consider residue classes modulo 4 since any player may play 4 minus what was played previously and thus preserve parity modulo 4. This will be referred to as *nimming down.* An extra property is that the number of ones and threes put down is the same, and the number of twos is even.

For $n \equiv 0\,(4)$ and $n \geq 8$, Player 2 should nim down until there is 8 left to play. From here, there are two cases.

(1) Player 1 is left to play on (E, E', E):
  - (i) If Player 1 puts down a 1 or a 3, then Player 2 puts down the other, leaving $(E+1, E', E+1)$.
    - (a) If Player 1 puts down a 1 or a 3, then Player 2 puts down the other, leaving player 1 to start on $(E+2, E', E+2)$, so Player 2 wins.
    - (b) If Player 1 puts down a 2, then Player 2 puts down a 1, forcing Player 1 to put down a 1. Player 2 then starts on $(E+3, E'+1, E+1)$, which is a win for Player 2.
  - (ii) If Player 1 puts a 2 down, Player 2 puts down a 1, leaving Player 1 to play on $(E+1, E'+1, E)$, with 5 total left to play.
    - (a) If Player 1 puts down a 1 or a 3, Player 2 puts down the other, forcing Player 1 to put down a 1. This leaves Player 2 to start on $(E+3, E'+1, E+1)$, which is a win for Player 2.
    - (b) If Player 1 puts down a 2, then Player 2 puts down a 2, and Player 1 is forced to put down a 1. Player 2 then starts on $(E+2, E'+3, E)$, which is a win for Player 2.

(2) Player 1 is left to play on (O, E, O):
  - (i) If Player 1 puts down a 1, then Player 2 puts down a 2, so Player 1 plays on $(O+1, E+1, O)$, with a total of 5 left to play.
    - (a) If Player 1 puts down a 1 or 3, then Player 2 puts down the other, forcing Player 1 to put down a 1. This leaves Player 2 to start on $(O+3, E+1, O+1)$, which is a win for Player 2.
    - (b) If Player 1 puts down a 2, then Player 2 puts down a 2, forcing Player 1 to put down a 1. Player 2 starts on $(O+2, E+3, O)$, which is a win for Player 2.
  - (ii) If Player 1 puts down a 2 or 3, then Player 2 puts down the other, so Player 1 plays on $(O, E+1, O+1)$ with 3 total left.
    - (a) If Player 1 puts down a 3, then Player 2 starts the game on $(O, E+1, O+2)$, which is a win for Player 2.
    - (b) If Player 1 puts down a 1 or 2, then Player 2 puts down the other. Player 1 then starts the game on $(O+1, E+2, O+1)$, which is a win for Player 2.

The other three cases are very similar. These, as well as the winning strategy for small values of n, have been included in the appendix. □

6. Further Directions and Conclusions

There are numerous directions for future work.

(i) While the authors believe Conjecture 1.3 is very much out of reach, Conjecture 1.2 can be proven by finding a family of Zeckendorf decompositions where Player 2 can force a win.

(ii) More generally, one might study other reversed games to see if any other games reveal such a complex winning structure upon reversal.

(iii) Future work could also try to solve the Reversed game for a larger family of starting positions, perhaps ones closer to the Zeckendorf decomposition.

(iv) Other modifications of the game can be considered, including a "stagnant one" variation, where all chips with value one are removed from the game (or equivalently, any move requiring a one cannot be played).

We conclude with a final, bold conjecture.

Conjecture 6.1. *The winning strategy for the forward Zeckendorf games is dependent upon who wins the reverse Zeckendorf game.*

This is a potential explanation of why a constructive proof of why Player 2 always wins the forwards game has remained elusive.

Appendix A. Additional Details

A.1. **Table of Wins for Reverse Zeckendorf Game.** The following table lists which player wins for the starting number n as well as the number of vertices and edges of the associated directed graph. A grayed square means Player 2 wins while a white one means Player 1 wins.

n	Result	Edges	Vertices
2		1	2
3	▨	2	3
4		4	4
5	▨	7	6
6		11	8
7		16	10
8	▨	24	14
9	▨	32	17
10		45	22
11	▨	59	27
12		77	33
13	▨	100	41
14	▨	126	49
15		158	59
16		198	71
17		241	83
18		297	99
19	▨	358	115
20	▨	430	134
21		516	157
22		610	180
23	▨	722	208
24		849	239
25		990	272
26		1158	312
27		1339	353
28		1548	400
29		1785	453
30		2043	509
31	▨	2341	573
32		2667	642
33		3028	717
34	▨	3440	803
35		3881	892
36		4381	993
37	▨	4930	1102
38		5528	1219
39	▨	6199	1350
40		6924	1489
41		7721	1640
42		8606	1808
43		9552	1983
44		10606	2178
45		11743	2386
46		12979	2609
47	▨	14339	2854
48		15796	3113
49	▨	17387	3393
50		19119	3697
51		20970	4017
52	▨	23001	4367
53	▨	25171	4737
54		27517	5134
55	▨	30064	5564
56	▨	32777	6016
57		35719	6504
58		38879	7025
59		42252	7575
60	▨	45909	8171
61	▨	49794	8797
62		53971	9466
63		58458	10183
64	▨	63222	10936
65		68351	11744

n	Result	Edges	Vertices
66	■	73811	12599
67		79627	13502
68		85874	14471
69		92480	15486
70	■	99552	16568
71		107083	17715
72		115060	18921
73	■	123593	20207
74	■	132622	21559
75		142212	22987
76	■	152437	24506
77		163215	26094
78		174701	27782
79	■	186852	29558
80		199697	31425
81	■	213356	33405
82		227747	35478
83		242990	37664
84	■	259136	39973
85		276121	42386
86		294140	44939
87		313108	47613
88	■	333117	50421
89	■	354284	53384
90	■	376512	56478
91		399993	59735
92	■	424730	63154
93		450710	66727
94		478155	70492
95		506942	74422
96		537240	78543
97	■	569148	82871
98	■	602577	87383
99		637787	92122
100		674725	97075
101		713458	102247
102	■	754214	107677
103	■	796847	113331
104		841620	119251
105		888582	125442
106		937699	131890
107		989279	138644
108	■	1043215	145681
109	■	1099679	153022
110		1158887	160703
111		1220686	168686
112	■	1285449	177031
113	■	1353148	185727
114	■	1423848	194777
115		1497885	204232
116		1575109	214059
117		1655815	224299
118		1740180	234978
119		1828092	246065
120	■	1920015	257632
121	■	2015842	269652
122		2115775	282150
123		2220148	295175
124	■	2328793	308687
125		2442154	322751
126		2560320	337374
127		2683290	352543
128	■	2811589	368337
129	■	2945040	384715

A.2. **Proof of Theorem 1.6.**

Theorem 1.6. *For any game starting with a ones, b twos, and c threes, we have the following forced wins.*

a	*b*	*c*		*Player having forced win*
Even	*Even*	*Even*		*Player 2*
Odd	*Odd*	*Odd*		*Player 1*
Even	*Odd*	*Even*		*Player 1*
Odd	*Even*	*Odd*		*Player 1*
Odd	*Even*	*Even*	$a > c$	*Player 2*
Odd	*Even*	*Even*	$a < c$	*Player 1*
Even	*Even*	*Odd*	$a > c$	*Player 1*
Even	*Even*	*Odd*	$a < c$	*Player 2*
Even	*Odd*	*Odd*		*Player 1*
Odd	*Odd*	*Even*		*Player 1*

Proof. For the (O, E, E) case, where $a < c$, we show Player 1 wins by force.

Player 1 wins by performing a combine with a one and a three, yielding the game state

$$(a-1, b+2, c-1). \tag{A.1}$$

Note $a-1 < c-1$. This also yields the (E, E, O) case with Player 2 on move. By work above, the player on move loses which means Player 1 wins.

For the (E, E, O) and $a > c$ case, we show Player 1 wins by force.

Player 1 wins by combining a one and a three. This gives the game state

$$(a-1, b+2, c-1). \tag{A.2}$$

Then, $a-1$ is odd, $b+2$ is even, and $c-1$ is even. Moreover, we have $a-1 > c-1$. Therefore, by the (O, E, E) proof, the player on move loses. Since Player 2 is on move, then Player 1 wins.

For the (E, E, O) and $a < c$ case, we show Player 2 wins by force.

First, assume Player 1 combines a one and a three. This gives the game state

$$(a-1, b+2, c-1). \tag{A.3}$$

Since Player 1 was able to combine a one and a three, then $a \geq 2$ which implies $c > 2$. We have $a-1, c-1 \geq 1$, and so Player 2 may perform another combine to get

$$(a-2, b+4, c-2). \tag{A.4}$$

The point is that $a-2 < c-2$ and all parities are preserved. Eventually, Player 1 will run out of ones and threes to combine and is forced to either split a three or a two. Both of these cases are covered next, and are wins for Player 2. (Informally, the main idea in this case is to force Player 1 to run out combines, upon which they will be forced to perform a split. We will show that all splits are losing for Player 1, and so Player 2 forces a win by running Player 1 out of combines.)

Assume Player 1 splits one of the twos. This gives the game state

$$(a+2, b-1, c). \tag{A.5}$$

Now, if $a+3 > c-1$, Player 2 wins by splitting one of the threes. (Since c is odd, a three exists.) This yields the game state

$$(a+3, b, c-1). \tag{A.6}$$

At this point, $a+3$ is odd, b is even, and $c-1$ is even. Moreover, we have $a+3 > c-1$. By work above, the player on move loses. Since Player 1 is on move, Player 2 wins.

On the other hand, if $a+3 < c-1$ (i.e., $a+4 < c$), Player 2 should copy Player 1 and split another two. Note that a two exists; since b is even and Player 1 was able to split a two, we must have $b \geq 2$. This forces $b-1 \geq 1$, and so a two is available. This yields the game state

$$(a+4, b-2, c). \tag{A.7}$$

The upshot is the number of ones is increasing, and so Player 2 will eventually be able to reduce to the situation analyzed in the $a+3 > c-1$ case. The parities are preserved and $a+4 < c$. Therefore, Player 2 wins.

Finally, assume Player 1 splits one of the threes. This yields the game state

$$(a+1, b+1, c-1). \tag{A.8}$$

If $c = 1$, then $c-1 = 0$. Since $a < c$, we get $a = 0$. The point is that we have the game state $(1, b+1, 0)$ with Player 2 on move. The only option for both players at this point is to keep splitting twos. Since $b+1$ is odd, Player 1 will run out of splits first, and so Player 2 wins.

Now, assume $c > 1$. If $a+3 > c-1$, Player 2 wins by splitting a two, yielding the position

$$(a+3, b, c-1). \tag{A.9}$$

Note $a+3$ is odd, b is even, and c is even. Since $a+3 > c-1$, then earlier work implies the player on move loses. As Player 1 is on move, Player 2 wins.

Finally, if $a+3 < c-1$, Player 2 wins by splitting a three, yielding the position

$$(a+2, b+2, c-2). \tag{A.10}$$

All parities are preserved, and $a+2 < c-2$ since $a+3 < c-1$. By induction, Player 2 wins.

In all cases, we have shown Player 2 has a forced win.

For the (E, O, O) case, we show Player 1 wins by force.

The winning strategy depends on the number of ones versus the number of threes. First, assume $a+2 > c$; that is, there is a large number of ones. Player 1 wins by splitting a three, yielding

$$(a+1, b+1, c-1). \tag{A.11}$$

Note $a+1$ is now odd, while $b+1$ and $c-1$ are even. Furthermore, we have $a+1 > c-1$ since $a+2 > c$. By earlier work, the player on move loses. Since Player 2 is on move, then Player 1 wins.

On the other hand, we assume $a+2 < c$; that is, there is a small number of ones. Player 1 wins by splitting a two, yielding the game state

$$(a+2, b-1, c). \tag{A.12}$$

Note $a+2$ is even, $b-1$ is even, and c is odd. Moreover, we have $a+2 < c$. By earlier work, the player on move loses. Since Player 2 is on move, then Player 1 wins.

In either case, we have exhibited a winning strategy for Player 1, and so Player 1 wins by force.

For the (O, O, E) case, we assume a and b are odd and c is even. We show Player 1 wins by force.

As before, the winning strategy depends on the number of ones versus the number of threes. First, assume $a+2 < c$. Player 1 wins by splitting a three, yielding

$$(a+1, b+1, c-1). \tag{A.13}$$

Note $a+1$ and $b+1$ are now even, while $c-1$ is odd. Furthermore, we have $a+1 > c-1$ due to $a+2 < c$. By earlier work, the player on move loses. As Player 2 is on move, Player 1 wins.

Finally, assume $a+2>c$. Player 1 wins by splitting a two, yielding the game state

$$(a+2, b-1, c). \tag{A.14}$$

Note $a+2$ is odd, $b-1$ is even, and c is even. Moreover, we have $a+2>c$. By earlier work, the player on move loses. As Player 2 is on move, Player 1 wins.

In either case, we have exhibited a winning strategy for Player 1, and so Player 1 wins by force. □

A.3. **Proof of Theorem 5.1.**

Theorem 5.1. *For $n \neq 4$,*

$$\begin{aligned} n = 4 &\implies \textit{Player 1 wins}, && (5.1)\\ n \textit{ odd} &\implies \textit{Player 1 wins}, && (5.2)\\ n \neq 4 \textit{ even} &\implies \textit{Player 2 wins}. && (5.3)\end{aligned}$$

Proof. For $n \equiv 1\ (4)$ and $n \geq 9$, Player 1 puts down a 3, and then nims down to 6. At this point, the board is either $(O, E, O+1)$ or $(E, E', E+1)$.

1. If Player 2 puts down a 1, the game state is either $(O+1, E, O+1)$ or $(E+1, E', E+1)$.
   - (i) In the case of $(O+1, E, O+1)$, Player 1 puts down a 2 leaving $(O+1, E+1, O+1)$.
     - (a) If Player 2 puts down a 1 or 2, then Player 1 puts down the other. Player 2 then starts on $(O+2, E+2, O+1)$, which is a win for Player 1.
     - (b) If Player 2 puts down a 3, then Player 1 starts on $(O+1, E+1, O+2)$ which is a win for Player 1.
   - (ii) In the case of $(E+1, E', E+1)$, Player 1 puts down a 3, leaving $(E+1, E', E+2)$.
     - (a) If Player 2 puts down a 1, then player 1 is forced to put down a 1. Player 2 then starts on $(E+3, E', E+2)$, which is a win for Player 1.
     - (b) If Player 2 puts down a 2, then Player 1 starts on $(E+1, E'+1, E+2)$, which is a win for Player 1.
2. If Player 2 puts down a 2 or 3, then Player 1 puts down the other, forcing Player 2 to put down a 1. Player 1 then starts on $(O+1, E+1, O+2)$ or $(E+1, E'+1, E+2)$, both of which are wins for Player 1.

For $n \equiv 2\ (4)$ and $n \geq 6$, Player 2 nims down until there is 6 left. The game state is either (O, E, O) or (E, E', E).

1. If Player 1 puts down a 1, then Player 2 is to move on $(O+1, E, O)$ or $(E+1, E', E)$, with 5 total left.
   - (i) In the case of $(O+1, E, O)$, Player 2 puts down a 3, leaving $(O+1, E, O+1)$, with a total of 2 left to play.
     - (a) If Player 1 puts down a 1, Player 2 is forced to put down a 1. Player 1 then starts on $(O+3, E, O+1)$, which is a win for Player 2.
     - (b) If Player 1 puts down a 2, Player 2 starts on $(O+1, E+1, O+1)$, which is a win for Player 2.
   - (ii) In the case of $(E+1, E', E)$, Player 2 puts down a 2, leaving $(E+1, E'+1, E)$.
     - (a) If Player 1 puts down a 1 or 2, then Player 2 puts the other down. Player 1 then starts on $(E+2, E'+2, E)$, which is a win for Player 2.
     - (b) If Player 1 puts down a 3, then Player 2 starts on $(E+1, E'+1, E+1)$, which is a win for Player 2.

(2) If Player 1 puts down a 2 or a 3, then Player 2 puts down the other, forcing Player 1 to play a 1. Player 2 then starts on $(O+1, E+1, O+1)$ or $(E+1, E'+1, E+1)$, both of which are wins for Player 2.

For $n \equiv 3\,(4)$, Player 1 puts down a 2, and then nims down to 1, upon which Player 2 puts down a 1. Player 1 then starts on an ordered triple (a, b, c) with b odd since nimming down preserves the parity of *twos* and there is a single 2 put down at the start. In all cases where b is odd, the player that starts wins, so Player 1 wins.

For $n = 1$, Player 1 puts down a 1. Player 2 then loses as they cannot move on $(1, 0, 0)$.

For $n = 2$, if Player 1 puts down a 1, Player 2 must do the same, and so Player 2 wins as Player 1 cannot move on $(2, 0, 0)$. If Player 2 puts down a 2, Player 2 starts on $(1, 1, 0)$, and therefore Player 2 wins.

For $n = 4$, Player 1 puts down a 3, forcing Player 2 to put down a 1. Player 1 starts on $(1, 0, 1)$, a winning position for Player 1.

For $n = 5$, Player 1 puts down a 2.

(i) If Player 2 puts down a 3, then Player 1 starts on $(0, 1, 1)$, a win for Player 1.
(ii) If Player 2 puts down a 1 or a 2, then Player 1 puts down the other. Player 2 then starts on $(1, 2, 0)$, which is a win for Player 1.

□

Appendix B. Code

The program we used to brute force solve who has a winning strategy is listed below. It was coded in Jupyter Notebook, but it can be run by any program that can run python. The computation complexity is about $O(\exp(\sqrt{n}))$, with the program taking about 2 hours for $n = 129$ and about 24 hours for $n = 144$.

```
import networkx as nx
from matplotlib import pyplot as plt
from networkx.drawing.nx_agraph import graphviz_layout

def is_game_a_loss(current_state):
    """
    Input: A list representing the current state of the game
    Output: Boolean

    Returns True if the game is over (i.e., is a
        loss for the player next to move)
    Returns False if the game is not over.
    """
    return sum(current_state[1:]) == 0


def combine(current_state):
    """
    Input: A list representing the current state of the game
    Output: A list of lists, where each list is a possible
        next state of the game

    Finds all the potential combine moves in the current
    state and creates the next states after the combine move
    """
    future_states = []
    for i, val in enumerate(current_state):
        tmp = list(current_state).copy()
        if i >= 3 and val and current_state[i-3]:
            tmp[i] -= 1
            tmp[i-3] -= 1
            tmp[i-1] += 2
            future_states.append(tuple(tmp))
    if len(current_state) > 2:
        if current_state[0] and current_state[2]:
            tmp = list(current_state).copy()
            tmp[0] -= 1
            tmp[2] -= 1
            tmp[1] += 2
            future_states.append(tuple(tmp))
    return set(future_states)

def split(current_state):
    """
    Input: A list representing the current state of the game
    Output: A list of lists, where each list is a possible next
        state of the game

    Finds all the potential split moves in the current state and creates
    the next states after the split move
    """

    if is_game_a_loss(current_state):
        return "An Error Has Occurred"
    future_states = []
    for i, val in enumerate(current_state):
        tmp = list(current_state).copy()
        if i == 0: continue
        if i == 1 and val:
            tmp[1] -= 1
            tmp[0] += 2
            future_states.append(tuple(tmp))
        elif val:
            tmp[i] -= 1
            tmp[i-1] += 1
            tmp[i-2] += 1
            future_states.append(tuple(tmp))
    return set(future_states)

def nearestSmallerEqFib(n):
    """
    Input: integer n
    Output: tuple, where the first entry is the greatest Fibonacci
        number smaller than n and the second is the index of
        that Fibonacci number
    """

    # Corner cases
    if (n == 0):
        return "404:BROKEN"
    elif n == 1:
        return (1,n)

    # Finds the greatest Fibonacci Number smaller
    # than n.
    f1, f2, f3 = 0, 1, 1
    index = 0
    while (f3 <= n):
        index += 1
        f1 = f2;
        f2 = f3;
        f3 = f1 + f2;
    return (f2, index);

def int_to_zeck(n):
    """
    Input: integer n
    Output: tuple that represents the Zeckendorf decomposition of n
    """
    arr = []

    while (n>0):
        f_i, i = nearestSmallerEqFib(n);
        arr.append(i)
        n = n-f_i

    arr2 = [0] * max(arr)
    for i in arr:
        arr2[i-1] += 1
    return tuple(arr2)

def Fibonacci(n):
    """
    Input: integer n
    Output: integer F_n, the n-th Fibonacci number (where F_2 = 2)
    """
    if n == 0:
        return 1
    # Check if n is 1,2
    # it will return 1
    elif n == 1:
        return 1

    else:
        return (Fibonacci(n-1) + Fibonacci(n-2))

def zeck_to_int(zeck_tuple):
    """
    Input: a tuple representing the Zeckendorf decomposition of a number n
    Output: the integer n
    """
    tot = 0
    for i, val in enumerate(zeck_tuple):
        tot += val * Fibonacci(i+1)

    return tot


def game_solver(number, draw_graph = False, graph_labels = False):
    """
    Input: an integer n (or a tuple representing a zeckendorf decomposition)
    Output: a tuple of the number, its zeckendorf decomposition, the winner,
        the number of edges, and vertices

    Also can draw the graph of the game
    """

    # Checks whether the input is a tuple or an integer, and
    # converts it to the zeckendorf decomposition if the latter
    if type(number) is tuple:
        initial_state = number
        number = zeck_to_int(number)
    else:
        initial_state = int_to_zeck(number)


    edges = []
    current_states = [initial_state]
    calculated_states = []

    # Continually loops through the list of possible
    # states to generate the next states for the game
    while True:
        future_states = []
        for current_state in current_states:
            if current_state in calculated_states:
                continue
            next_states = set()
            if is_game_a_loss(current_state): continue
            next_states = next_states.union(split(current_state))
            next_states = next_states.union(combine(current_state))
            edges += ((current_state, state) for state in next_states)
            future_states += next_states.copy()
            calculated_states += [current_state]
        future_states = set(future_states)

        # If there are no states left to be calculated,
        # we must be at the end node, i.e., game is over
        if len(future_states) == 0:
            break
        current_states = future_states.copy()

    # Initializes the graph of the game
    G = nx.DiGraph()
    for edge in set(edges):
        G.add_edge(edge[0], edge[1])

    # Sets the initial win/loss state for the vertices, all are set to False
    # (i.e., not calculated) except the end node, which is a loss
    data = dict()
    for node in G.nodes:
        if is_game_a_loss(node):
            data[node] = "L"
        else:
            data[node] = False

    """
    Goes through the game tree in reverse
    and calculates the state of each vertex

    If any vertex leads to one that is a loss,
        that vertex is automatically a win

    If all the child nodes of a vertex are a win,
        then that vertex is a loss
    """
    not_done = True
    while not_done:
        for node in G.nodes:
            if data[node]: continue
            children_states_L = [data[node] == "L"
                        for node in G.neighbors(node)]
            children_states_W = [data[node] == "W"
                        for node in G.neighbors(node)]
            if any(children_states_L):
                data[node] = "W"
            elif all(children_states_W):
                data[node] = "L"

        not_done = not all(data[node] for node in G.nodes)

    # Colors the vertices based on whether they are wins or losses
    color_map = []
    for node in G.nodes:
        if sum(node[1:]) == 0:
            color_map.append("black")
        elif data[node] == "W":
            color_map.append("green")
        elif data[node] == "L":
            color_map.append("red")
        if node == initial_state:
            if data[node] == 'W': winner = 1
            else: winner = 2

    # Draws the graph
    if draw_graph:
        plt.figure(3,figsize=(20,20))
        pos=graphviz_layout(G, prog='dot')
        nx.draw_networkx(G, pos, node_color=color_map, with_labels=graph_labels)

    return (number, "".join([str(i) for i in initial_state]),
        winner, len(G.edges), len(G.nodes))

# Example: should output a 5-tuple (7, '0101', 1, 16, 10)
#   and a graph representing the game
game_solver(7, True, True)
```

References

[BCDD+15] E. Boldyriew, A. Cusenza, L. Dai, P. Ding, A. Dunkelberg, J. Haviland, K. Huffman, D. Ke, D. Kleber, J. Kuretski, J. Lentfer, T. Luo, S. J. Miller, C. Mizgerd, V. Tiwari, J. Ye, Y. Zhang, X. Zheng, and W. Zhu, *Extending Zeckendorf's Theorem to a Non-constant Recurrence and the Zeckendorf Game on this Non-constant Recurrence Relation*, Fibonacci Quarterly **58** (2020), no. 5, 55–76.

[BEFM1] P. Baird-Smith, A. Epstein, K. Flint and S. J. Miller, *The Zeckendorf Game*, Combinatorial and Additive Number Theory III, CANT, New York, USA, 2017 and 2018, Springer Proceedings in Mathematics & Statistics **297** (2020), 25–38.

[BEFM2] P. Baird-Smith, A. Epstein, K. Flint and S. J. Miller, *The Generalized Zeckendorf Game*, Proceedings of the 18th International Conference on Fibonacci Numbers and Their Applications, Fibonacci Quarterly **57** (2019), no. 5, 1–14.

[CDHK+6i] A. Cusenza, A. Dunkelberg, K. Huffman, D. Ke, D. Kleber, S. J. Miller, C. Mizgerd, V. Tiwari, J. Ye and X. Zheng, *Winning Strategy for the Multiplayer and Multialliance Zeckendorf Games*, Fibonacci Quarterly **59** (2021), 308–318.

[CDHK+6ii] A. Cusenza, A. Dunkelberg, K. Huffman, D. Ke, D. Kleber, M. McClatchey, S. J. Miller, C. Mizgerd, V. Tiwari, J. Ye, X. Zheng, *Bounds on Zeckendorf Games*, Fibonacci Quarterly **60** (2022), no. 1, 57–71.

[CMJDMMN] Justin Cheigh, Guilherme Zeus Dantas E Moura, Ryan Jeong, Jacob Lehmann Duke, Wyatt Milgrim, Steven J. Miller, Prakod Ngamlamai, *Towards The Gaussianity Of Random Zeckendorf Games*, preprint. `https://arxiv.org/abs/2210.11038`

[GMRVY] Diego Garcia-Fernandezsesma, Steven J. Miller, Thomas Rascon, Risa Vandegrift, and Ajmain Yamin, *The Accelerated Zeckendorf Game*, to appear in Fibonacci Quarterly

[LLMMSXZ] R. Li, X. Li, S. J. Miller, C. Mizgerd, C. Sun, D. Xia, Z. Zhou, *Deterministic Zeckendorf Games*, Fibonacci Quaterly **58** (2020), no. 5, 152-160.

[MSY] Steven J. Miller, Eliel Sosis, and Jingkai Ye, *Winning Strategies for the Generalized Zeckendorf Game*, Fibonacci Quaterly, Conference Proceedings: 20th International Fibonacci Conference: **60** (2022), no.5, 270–292

[Ze] E. Zeckendorf, Représentation des nombres naturels par une somme des nombres de Fibonacci ou de nombres de Lucas, Bulletin de la Société Royale des Sciences de Liège **41** (1972), pages 179–182.

MSC2020: 11B39, 65Q30, 05C57, 91A05, 91A06

Pomona College
Email address: zxba2020@mymail.pomona.edu

University of Cambridge
Email address: aj644@cam.ac.uk

Williams College
Email address: sjm1@williams.edu, Steven.Miller.MC.96@aya.yale.edu

University of Michigan
Email address: anaray@umich.edu

University of Cambridge
Email address: kds43@cam.ac.uk

University of Cambridge
Email address: aky30@cam.ac.uk

Yale University
Email address: chris.yao@yale.edu