Rose-Hulman Undergraduate Mathematics Journal

Volume 14
Issue 2

Article 3

The Broncho Tower: A Modification of the Tower of Hanoi Puzzle

Stephen B. Gregg
University of Central Oklahoma, stephen.b.gregg@gmail.com

Juan Orozco
University of Central Oklahoma, jorozco@uco.edu

Rose-Hulman Undergraduate Mathematics Journal

The Broncho Tower: A Modification of the Tower of Hanoi Puzzle

Stephen B. Gregg[a] Juan Orozco[b]

Volume 14, no. 2, Fall 2013

Sponsored by

Rose-Hulman Institute of Technology
Department of Mathematics
Terre Haute, IN 47803
Email: mathjournal@rose-hulman.edu
http://www.rose-hulman.edu/mathjournal

[a]University of Central Oklahoma
[b]University of Central Oklahoma

Rose-Hulman Undergraduate Mathematics Journal
Volume 14, no. 2, Fall 2013

The Broncho Tower: A Modification of the Tower of Hanoi Puzzle

Stephen B. Gregg Juan Orozco

Abstract. We explore ways to find an upper bound on the minimum number of moves to complete a modified version of the Tower of Hanoi puzzle. First, we demonstrate how a simpler modification's minimum number of moves can be obtained using Difference Equations. Next, we show how our modified puzzle differs and how we can apply the same techniques to our puzzle.

Acknowledgements: Special thanks to Johnny Sharp for his work on the algorithms used in the computer program. We would also like to thank our advisors, Britney Hopkins, Kristi Karber, and Thomas Milligan.

1 Introduction

The Tower of Hanoi puzzle is a mathematical game that consists of n disks and three pegs, which was made famous by Édouard Lucas in 1883 [3]. Lucas, who also studied the Fibonacci sequence, discovered the Mersenne prime $2^{127} - 1$. This is the largest Mersenne prime discovered without the use of a computer [4]. The Fibonacci sequence is of relevance to the Tower of Hanoi because both the Fibonacci sequence and the Tower of Hanoi puzzle involve recursive relationships. The Tower of Hanoi puzzle begins with n disks, graduated in size, arranged on the first peg in descending order from the bottom to the top. Disks can only be moved one at a time, and a disk cannot be placed on top of a smaller disk. After the disks have all been moved and arranged in descending order on the third peg, the game is completed. There are many ways to complete the puzzle, but an optimal solution is only achieved when the least number of moves are made. The minimum number of moves that it takes to complete the Tower of Hanoi puzzle with n disks is $2^n - 1$. Notice here that Mersenne primes are of the form $2^n - 1$, so studying these numbers is one possible motivation for the interest Lucas had in the Tower of Hanoi puzzle.

In [6], John McCarthy, from the University of Stanford, proposed a modified version of the Tower of Hanoi. With this version, called the Tower of Stanford puzzle, a disk can be placed on top of a smaller disk provided the disk at the bottom of that stack is the largest on that peg. As with the Tower of Hanoi puzzle, the goal is to find the least number of moves that it takes to complete this puzzle. Lee Badger and Marc Williams, from Weber State University, provided a solution of $n^2 - n + 1$, which gives the optimal solution to complete the Tower of Stanford with n disks [7].

We have made a stricter modification to the original Tower of Hanoi puzzle and we have called it the Broncho Tower puzzle. With the Broncho Tower, we changed the rule preventing a disk from being moved on top of a smaller one. In our puzzle, a disk can be moved on top of a smaller one provided that the two disks on the bottom of that stack are the largest on that peg (with the largest disk on the bottom and the second largest disk on top of that one). In an effort to find the optimal solution to the Broncho Tower puzzle, we have analyzed the Tower of Stanford puzzle in a Difference Equations context. This has allowed us to view their puzzle as several smaller processes. Using this, we have found a difference equation that models one aspect of the Broncho Tower puzzle and accounts for our stricter rules. The solution to this difference equation together with some additional moves (which we will cover later), gives an upper bound for an optimal solution to the Broncho Tower puzzle.

In Section 2, we provide the rules for the Tower of Hanoi and the Tower of Stanford puzzles. We also give an alternate proof for the optimal solution for the Tower of Stanford puzzle using Difference Equations. In Section 3, we provide the rules for the Broncho Tower puzzle, and we state and prove an upper bound on the minimum number of moves that it takes to complete it. Finally, in Section 4, we discuss some additional modifications that can be made to the Tower of Hanoi puzzle.

2 Preliminary

The Tower of Hanoi puzzle has the strictest rules of the different puzzles that we consider. That is, there are no cases when special moves can be made. The following rules for the Tower of Hanoi will later be modified to produce the different variations on the puzzle.

Rules for the Tower of Hanoi Puzzle

- Disks can only be moved one at a time
- Only the top disk on a stack can be moved from that peg
- A disk cannot be placed on top of a smaller disk

When playing the Tower of Hanoi puzzle, one finds that in order to reach the end goal, there are lesser goals that must be achieved first. For example, if one starts with a stack of seven disks, he or she must first go through a sequence of moves to move the smallest six of those to the second peg so that the largest disk can then be placed on the third peg. Then, to move those six disks on top of the largest disk, he or she must first make moves to move the smallest five of those to the first peg to free up the bottom disk. These recursive relationships are used to construct the model of the puzzle. Now we are going to look at the Tower of Stanford puzzle to see how a change to the rules affects the way that it is completed.

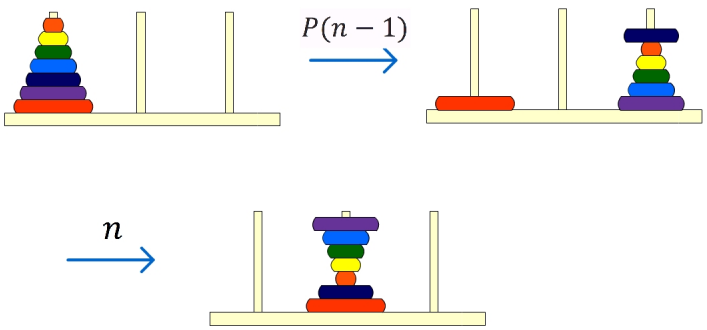
Rules for the Tower of Stanford Puzzle

- Disks can only be moved one at a time
- Only the top disk on a stack can be moved from that peg
- *A disk can be placed on top of a smaller disk provided the disk at the bottom of that stack is the largest on that peg*

We prove the theorem below using techniques from Difference Equations.

Theorem. *The Tower of Stanford puzzle with n disks has an optimal solution of $n^2 - n + 1$.*

Proof. Let $P(n)$ be the minimum number of moves to move n disks from one peg to another peg without regard to their order.



After moving the first $n-1$ disks, we can stack the remaining disks on top of these, giving $P(n) = P(n-1)+n$. Now, since it takes only one move to move one disk, we have $P(1) = 1$. This gives us a difference equation

$$P(n+1) - P(n) = n+1, \quad P(1) = 1.$$

Note that the falling factorial for $r \in \mathbb{Z}^+$ is defined by $t^{\underline{r}} = t(t-1)(t-2)\cdots(t-r+1)$.

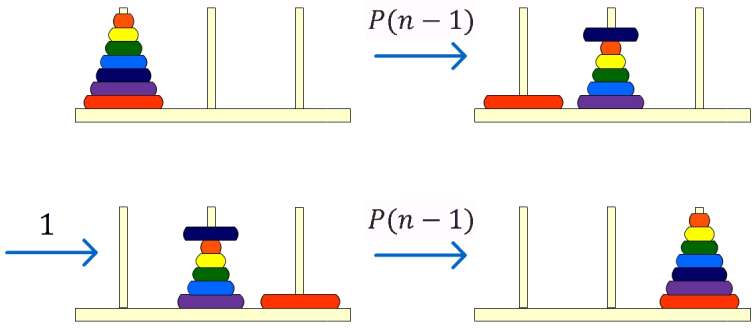
Now, $\Delta P(n) = n+1$. Taking the indefinite sum of both sides, we get

$$P(n) = \sum (n+1) = \sum (n^{\underline{1}} + 1) = \frac{n^{\underline{2}}}{2} + n + C = \frac{n(n-1)}{2} + n + C.$$

Since $P(1) = 1$, $C = 0$ and we get

$$P(n) = \frac{n(n+1)}{2}.$$

Next, let $R(n)$ be the minimum number of moves required to complete the Tower of Stanford puzzle.



It will take $P(n-1)$ moves to move the first $n-1$ disks onto the second peg, 1 more move to move the largest disk onto the third peg, and another $P(n-1)$ moves to move the $n-1$ disks back onto the largest disk on the third peg. Thus, $R(n) = 2P(n-1)+1$. Now, since $P(n) = \frac{n(n+1)}{2}$, we have $R(n) = 2\left(\frac{(n-1)n}{2}\right)+1$ or $R(n) = n^2 - n + 1$. □

3 Results

The Broncho Tower puzzle involves a stricter modification of the third rule.

Rules for the Broncho Tower Puzzle

- Disks can only be moved one at a time
- Only the top disk on a stack can be moved from that peg
- *A disk can be moved on top of a smaller one provided that the two disks on the bottom of that stack are the largest on that peg (with the largest disk on the bottom and the second largest disk on top of that one)*

As the number of disks increases, the number of possible moves that one can make while attempting to complete the Broncho Tower puzzle becomes large. Also, this version of the game is not as intuitive to play as the Tower of Hanoi puzzle, making it more difficult to find the quickest route to complete it. This difficulty is the result of having multiple options for where disks can be moved throughout the puzzle. So to help us find the quickest route to the end of the puzzle, we have written a computer program in C++ to try all of the different ways to finish the puzzle. We start by placing an upper bound on the minimum number of moves that it takes to complete the Broncho Tower puzzle. This bound is equal to the optimal solution to the Tower of Hanoi puzzle. We use iteration in the computer program to find routes taking a number of moves less than or equal this bound. Then we pick the route that completed the Broncho Tower puzzle in the fewest moves and use that as the new upper bound on the number of moves required to complete the puzzle for that number of disks.

The upper bound, $T(n)$, for the minimum number of moves required to complete the Broncho Tower puzzle with n disks is

$$T(n) = \begin{cases} 2^n - 1 & \text{if } n = 1, 2, 3, 4 \\ 29 & \text{if } n = 5 \\ 2n^2 - 4n + 1 & \text{if } n \geq 6. \end{cases}$$

Notice that for 1, 2, 3 and 4 disks, the minimum number of moves given by our upper bound is the same minimum for the Tower of Hanoi puzzle. Thus there is no change in how

the puzzle is completed in these cases. For 5 disks, our upper bound is two fewer moves than the optimal solution to the Tower of Hanoi puzzle. There are only two steps where a larger disk can be placed on top of a smaller disk, otherwise the puzzle is played the same as the Tower of Hanoi for 5 disks. These two steps reduce the minimum number of moves for the Broncho Tower puzzle to 29 as opposed to 31. As the number of disks increases beyond 5, the modified rules of the Broncho Tower start to allow for more opportunities to place larger disks on top of smaller ones. However, these instances follow a pattern and the following theorem gives an upper bound for the minimum number of moves required to complete the Broncho Tower for 6 or more disks.

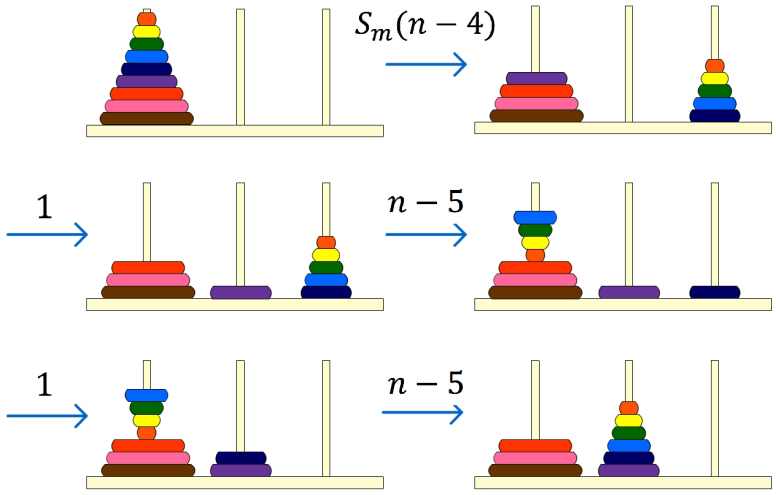
Theorem. *The optimal solution for the Broncho Tower puzzle on $n \geq 6$ disks has an upper bound of $T(n) = 2n^2 - 4n + 1$.*

Proof. In order to model our game, we split the completion of it into three processes that we will consider separately.

Process 1:

For this step we will only be concerned with moving the smallest $n-3$ disks from the first peg onto the second peg in the least number of moves. The reason we have chosen to do this is because the three largest disks will always be on the first peg, so we will always have a peg to move disks onto without breaking our rules. This allows us to stack the other disks on this peg in any order we want since they are smaller than these three largest ones. Depending on where the largest disks are located throughout the puzzle, we will achieve fewer or greater number of *saves*. In this case a save is a special move that the modified rules have allowed us to make that progresses the completion of the puzzle more quickly than the moves made in the classic Tower of Hanoi puzzle. Some saves benefit us more than other saves, but we do not always take the saving moves because they sometimes end up costing us in the end. We will see this concept in greater detail in the second process.

Let $S_m(n)$ be the minimum number of moves required to move n disks from a peg to another peg without regard to their order. Here we consider $S_m(n)$ for $n = 9$ disks to illustrate the concept.



In order to move the smallest $n-3$ disks, we first move the smallest $n-4$ disks using the $S_m(n)$ function. So we have that

$$S_m(n-3) = S_m(n-4) + 1 + (n-5) + 1 + (n-5) = S_m(n-4) + 2n - 8.$$

Also, it takes only one move to move one disk so we have $S_m(1) = 1$. Applying the shift operator and rewriting it as a forward difference, we have

$$\Delta S_m(n) = 2n, \quad S_m(1) = 1.$$

Taking the indefinite sum of both sides, we get

$$S_m(n) = \sum 2n^{\underline{1}} = 2(\frac{n^{\underline{2}}}{2}) + C = n^{\underline{2}} + C.$$

Now using our initial condition to solve for C, we get

$$S_m(n) = n(n-1) + 1 = n^2 - n + 1.$$

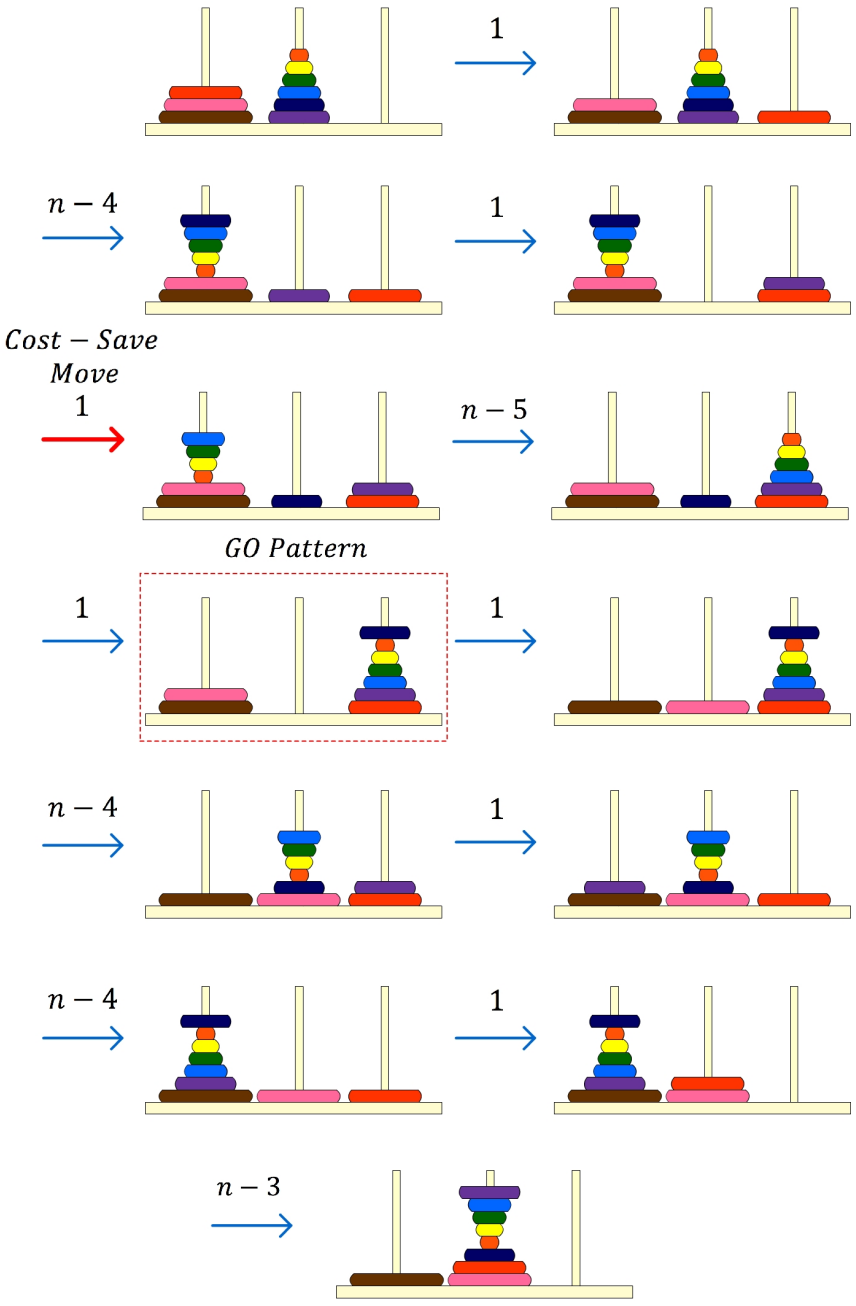
Thus to move the smallest $n-3$ disks, it will take

$$S_m(n-3) = (n-3)^2 - (n-3) + 1 = n^2 - 7n + 13$$

moves. This completes Process 1 and when we begin Process 2, we will start with the disks arranged as they ended in Process 1.

Process 2:

After the completion of this step, we will be at the halfway point for completing the puzzle. Once that is achieved we will be able to reverse the steps. Unlike the previous process, this process consists of 12 distinct sets of moves (7 of which are single moves and 5 of which are based on the starting number of disks) and does not involve a difference equation. Once again, we illustrate this conceptually using $n = 9$ disks.

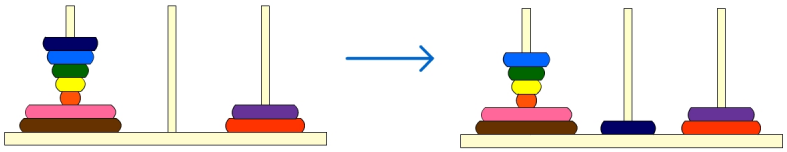


Process 1 freed up the third peg so that we can now move the third largest disk onto it. This is the first move made in Process 2. We still have the two largest disks on the first peg and we need to get the $n-4$ smallest disks on top of them to free the disk beneath. This takes $n-4$ moves. Then we move the fourth largest disk on top of the third largest. The next move is not an intuitive move, but it is a necessary move in order to achieve an overall optimal solution. This unique move is called the *Cost-Save move*; it is the placing of the

fifth largest disk onto the empty second peg. After this, the $n-5$ smallest disks on the first peg are placed on the third peg. This takes $n-5$ moves. Next, the fifth largest disk (the disk last moved by the Cost-Save move) is placed on top of the stack of disks on the third peg, giving us the GO pattern, which we address below. Then the second largest disk is placed on the second peg. Next we take the top $n-4$ disks from the third peg and place them on the second peg. This takes $n-4$ moves. Notice that because of the Cost-Save move we were able to stack the disks directly onto the second peg. This is because the bottom two disks were larger than all of the ones we wanted to move. Next, we move the fourth largest disk onto the first peg. Then we move the top $n-4$ disks from the second peg onto the first peg. This takes $n-4$ moves. Now we move the third largest disk onto the second peg. Our final set of moves for Process 2 is to take all of the disks from the first peg, except the bottom one, and move them onto the second peg. Since our second and third largest disks are on the second peg already, this only takes $n-3$ moves, getting us to the halfway point. Adding up all of these moves, we get

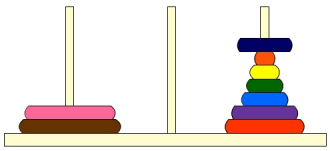
$$1+(n-4)+1+1+(n-5)+1+1+(n-4)+1+(n-4)+1+(n-3)=5n-13.$$

We have now completed Process 2, and from it we have identified an important configuration that appears while completing the puzzle for any number of disks. We are led into the GO configuration by means of the Cost-Save move.



(The Cost-Save Move for 9 disks)

The Cost-Save move occurs whenever the third and fourth largest disks are by themselves on the third peg, and the first peg contains the other disks arranged in the following order from the bottom: largest, second largest, the remaining disks in ascending order. This move is not intuitive because if the move was not made, that is, if the fifth largest disk was placed on the third peg, the two largest disks on the first peg would be freed in fewer moves. However, taking the seemingly quicker of the two routes would take more moves to free up the largest disk. A cost of one move is made so that the maximum number of saves can be achieved. This shows that the minimum number of moves required to complete the puzzle is not the sum of the minimum number of moves to free up each individual disk.



(The GO pattern for 9 disks)

Whenever the largest and second largest disks are on the first peg, it is a configuration we call the *Gregg-Orozco (GO) pattern*. This appears to be the optimal pattern (as seen in our computer program) leading up to the moving of the second largest disk onto the second peg. Having the disks in this configuration results in the most saves overall. We could spend fewer moves to stack disks up, but when we unstack them, it would take too many moves. Likewise, we could find a configuration that took very few moves to unstack, but getting to that configuration would take too many moves. The GO pattern quantifies the best way to stack disks so that when they have to be unstacked, the fewest number of moves are used for the two tasks done together.

Process 3:

The final part of the Broncho Tower puzzle is to move the largest disk onto the third peg and then repeat Process 1 and Process 2 in reverse order. Let $T(n)$ be our upper bound for the optimal solution for the Broncho Tower puzzle. Then, for $n \geq 6$,

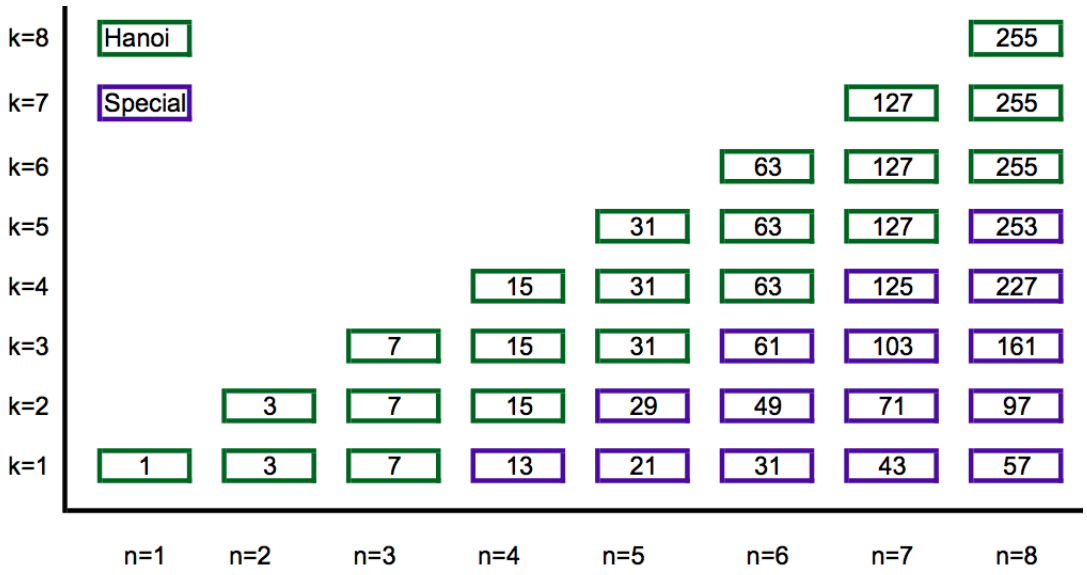
$$T(n) = (n^2 - 7n + 13) + (5n - 13) + 1 + (n^2 - 7n + 13) + (5n - 13) = 2n^2 - 4n + 1.$$

□

4 Further Work

While we believe our upper bound does give the optimal solution, we have yet to prove that this bound is sharp.

For the Tower of Stanford, the bottom disk must always be the largest in any stack of disks and for the Broncho Tower, the bottom two disks must always be the largest in any stack of disks. If we introduce a value called k and let it equal the number of disks in any given stack that must always be the largest, k would equal one for the Tower of Stanford and k would equal two for the Broncho Tower. We would like to continue finding upper bounds for larger values of k. Also, we would like to find a general formula for the upper bound for a puzzle with an arbitrary number of disks and an arbitrary value of k. Our program has been designed to find solutions for various values of k with n disks, and the following chart shows the minimum number of moves required to complete the puzzle for a few of those values.



The green colored boxes represent values of k and n whose corresponding optimal solutions do not differ from the Tower of Hanoi puzzle. For example, with $n = 6$ disks, it takes just as many moves to complete the puzzle if we require the bottom 4 disks ($k = 4$) to always be the largest in a stack, as it does if we require the bottom 6 disks to always be the largest in a stack ($k = 6$). The purple colored boxes represent modifications that are played differently than the Tower of Hanoi puzzle.

References

[1] Gillman, R., *Everyday Questions, Not-So-Everyday Mathematics*, The Mathematical Association of America 74th Annual Meeting of the Oklahoma-Arkansas Section, Arkadelphia, Arkansas, March 30-31, 2012.

[2] Kelley, W.G., Peterson A.C., *Difference Equations: An Introduction with Applications*, Academic Press, Salt Lake City, 2000.

[3] Lucas, É., Récréations Mathématiques, Vol. III, Gauthier-Villars, Paris, 1893. Reprinted several times by Albert Blanchard, Paris.

[4] MacTutor History of Mathematics archive. *François Edouard Anatole Lucas*, December 1996.
`http://www-groups.dcs.st-and.ac.uk/history/Biographies/Lucas.html`

[5] Sharp, J.C., Private communications, October 2012-April 2013.

[6] The Tower of Stanford, *Problems and Solutions*, American Mathematical Monthly **Vol. 109, Number 7** (August-September 2002), 664.

[7] The Tower of Stanford, *Problems and Solutions*, American Mathematical Monthly **Vol. 111, Number 4** (April 2004), 364–365.